

Handelshögskolan vid Göteborgs universitet
Institutionen för informatik

RIKTLINJER VID DESIGN AV ANVÄNDARGRÄNSSNITT

SAMT

INTERNET EXPLORER VS NETSCAPE NAVIGATOR - EN STUDIE I DESIGN OCH ANVÄNDARVÄNLIGHET

"Har du tänkt på hur användarna hanterar de tjusiga gränssnitt du designar? Visst, de klickar på knapparna och länkar med länkarna, men hur använder de det egentligen? Gör de som du hade tänkt dig?

Får de det resultat som de hade tänkt sig? Kan de utföra de arbetsuppgifter de hade tänkt sig? Eller kan de bara göra det du hade tänkt dig? Och hade du i så fall tänkt rätt?"

Ur tidningen Datateknik nr 29, 1998

Frågor som varje designer av användargränssnitt borde ställa sig, men vilket tyvärr inte alltid sker. I denna rapport, som delats upp i två huvudavsnitt, förutom ett inledande kapitel om interaktion mellan människa och dator, formuleras ett antal riktlinjer som bör övervägas vid utformning av ett användarvänligt och funktionellt gränssnitt. Det har framkommit att det inte bara är användning och placering av visuella komponenter, såsom fönster, menyer och kontroller som har betydelse, utan även slutanvändarens mentala kapacitet och fysiska omgivning. I en fallstudie jämförs de ledande webbläsarna Internet Explorer och Netscape Navigator mot de framtagna riktlinjerna. Webbläsare är till för att på ett så enkelt och effektivt sätt kunna förflytta sig på nätet och att skicka mail. Avancerade funktioner har svårt att komma till sin rätt, då användare ej tar sig tid att utforska dem om de ens upptäcks.

Examensarbete 10p för ADB-programmet 80p
Vårterminen 1999
Handledare: Roy Corneliusson

Jessica Hellström & Maria D. Nilsson

INNEHÅLLSFÖRTECKNING

INTRODUKTION.....	2
BAKGRUND	2
SYFTE OCH AVGRÄNSNING	2
DISPOSITION	3
SPRÅKBRUK	3
SAMSPELET MELLAN MÄNNISKA OCH DATOR.....	4
VARFÖR SKALL ANVÄNDAREN INVOLVERAS I DESIGNPROCESSEN?.....	4
VAD VÄNTAR OSS I FRAMTIDEN?	5
ATT DESIGNA ETT ANVÄNDARGRÄNSSNITT	6
INTRODUKTION	6
<i>Syfte & metod</i>	6
ANVÄNDARNA	7
<i>Låt oss tänka på saken!</i>	7
<i>Erfarna vs ovana användare</i>	8
<i>Det är inte bara insidan som räknas</i>	9
DET GRAFISKA ANVÄNDARGRÄNSSNITTET HAR ERÖVRAT VÄRLDEN	9
<i>Direkt manipulation</i>	10
GRUNDLÄGGANDE PRINCIPER VID DESIGN AV ANVÄNDARGRÄNSSNITT	11
<i>Allmänna riktlinjer</i>	12
<i>Layout</i>	15
Fönster	15
Verktygsfält & paletter	17
Menyer.....	18
Kontroller.....	20
Färger	26
SAMMANFATTNING	28
INTERNET EXPLORER VS NETSCAPE NAVIGATOR	30
INTRODUKTION	30
<i>Syfte</i>	30
<i>Utvärderingsmetoder</i>	30
BAKGRUND	31
UNDERSÖKNING	32
<i>Design</i>	33
<i>Användarvänlighet</i>	36
RESULTAT.....	40
<i>Design</i>	41
<i>Användarvänlighet</i>	42
DISKUSSION	44
VAD VI HADE KUNNAT GÖRA ANNORLUNDA	44
VAD VI TYCKER OCH TÄNKER OM ÄMNET	44
VAD VI KAN FÖRVÄNTA OSS I FRAMTIDEN.....	45
REFERENSLISTA	46

INTRODUKTION

Bakgrund

För inte allt för många år sedan, var datorer stora, komplicerade maskiner som inte bara krävde mycket elektricitet, utan också ständig tillsyn av experter. För att på den tiden bli en sk *datoranvändare*, krävdes inte bara utbildning och kunskap om kommandon och koder för att få datorerna att utföra önskade operationer, utan även labb-rock.

Idag har datorn en självskriven plats på våra kontor, och hemanvändningen ökar hela tiden kraftigt. Den information som tidigare lästes från pappersbaserade dokument, samlade i pärmar eller utspridda på skrivbordet, tas numera från det lilla fönster som utgör bildskärmen. Pärmar har ersatts av mappar i form av ikoner och att bläddra sig igenom buntar av papper består numera av att manipulera mer eller mindre strukturerade användargränssnitt.

Sättet på hur dessa användargränssnitt är designade och hur information presenteras på skärmen har stor betydelse för hur väl användaren kan utföra sina uppgifter. Under våra två år på ADB-programmet har vi sett många exempel på hur användargränssnitt *inte* ska designas. Det verkar finnas ett stort behov av mer kunskap och insikt inom detta område både på institutionen och ute i näringslivet. Därför ansåg vi att detta var ett lämpligt ämne att fördjupa sig i, samtidigt som det egna intresset var stort.

Syfte och avgränsning

Vi vill med denna rapport öka förståelsen för hur viktig användargränssnittsdesign är och formulera ett antal riktlinjer för vad som bör tänkas på vid skapandet av ett funktionellt och användarvänligt gränssnitt.

Efter att ha tittat på olika författares teorier om hur användargränssnitt på bästa sätt bör designas, har vi kommit fram till att de har ungefär samma tankar och idéer i ämnet. Därför har vi valt att inte jämföra olika teorier, utan istället sammanställt de riktlinjer vi anser vara viktigast. Vi har ej heller tagit upp några tekniska aspekter på gränssnitt.

Studien i del två syftar till att se hur pass väl de framtagna riktlinjerna följs i verkligheten. Anledningen till att vi har valt Netscape Navigator och Internet Explorer är att dessa är de två vanligaste webbläsarna idag, och därför bör följa de riktlinjer som leder till god design. För att begränsa vår undersökning har vi valt att helt utelämna mail-funktionerna och övriga tilläggsfunktioner, som exempelvis Netscapes Composer.

Uppsatsen riktar sig till de studenter som sysslar med systemutveckling, där design av användargränssnitt ofta helt glöms bort, eller är ett ”nödvändigt ont” som först blir aktuellt om tiden räcker till...

Vi har valt att skriva vår uppsats på ett sätt så att den ska vara förståelig för alla med något intresse för ämnet. Vi har medvetet undvikit att uttrycka oss i alltför tekniska termer eftersom vi vill att uppsatsen ska vara rolig att läsa och inspirera till att riktlinjerna används som grund i gränssnittsdesign, eller åtminstone att det leder till eftertanke innan en mörkgrön bakgrund används till en gul text - ett exempel som vi tyvärr själva råkat stöta på!

Disposition

Arbetet är, förutom en inledande beskrivning av samspelet mellan människa och dator, uppdelat i två delar. I den första delen behandlar vi design av användargränssnitt i allmänhet. Vi betonar hur viktigt det är med användarmedverkan, berättar om grafiska användargränssnitt, samt sammanställer viktiga riktlinjer vid design.

I del två fokuserar vi på webbdesign. Det inleds med en kort historia om webbläsare och sedan görs en undersökning i design och användarvänlighet med de två största webbläsarna idag - Netscape Navigator och Internet Explorer.

Språkbruk

Eftersom vi de senaste åren bombarderats med nya datatermer på engelska som många saknar svensk översättning, har det resulterat i att flera av våra uttryck i uppsatsen inte alltid stämmer överens med vad som brukar kallas ”god svenska”. I vissa fall har vi helt enkelt använt oss av det engelska uttrycket, då det ej har funnits en tillfredsställande översättning.

SAMSPELET MELLAN MÄNNISKA OCH DATOR

Det har forskats en hel del kring samspelet mellan människa och dator. Det är ett område som innefattar hur människor och datorer kommunicerar eller interagerar med varandra, och även hur användaren både fysiskt och psykiskt uppfattar sin omgivning. Det är den engelska termen ***Human Computer Interface***, med förkortningen ***HCI***, som kommit att stå som rubrik för dessa studier, och det är även den termen som fortsättningsvis kommer att användas här.

Ett system kan delas in i två nivåer: ***systemets tjänster***, dvs vad systemet kan prestera, och ***användargränssnittet***. Ett systems tjänster bestämmer ***vad*** användaren kan utföra med det, och användargränssnittet bestämmer ***hur*** det kan utföras, dvs den del av programmet som kontrollerar hur information presenteras för användaren och hur datorn tar emot input från dennes. Om tjänsterna är fel, är systemet fel. Om tjänsterna är rätt, då måste användargränssnittet designas så att användaren kan utnyttja dessa på ett optimalt sätt. Detta är vad HCI handlar om.

Förut låg ansvaret hos användaren att förse systemet med rätt form av input, vilket krävde lång utbildning och träning. Numera har det ansvaret flyttats över till designern av systemet. Denne måste se till att göra systemet mer formbart efter hur människor arbetar och kommunicerar på ett naturligt sätt. Hela idén är att system ska designas till att stödja användaren på de områden, där de mänskliga egenskaperna inte räcker till.

Varför skall användaren involveras i designprocessen?

I all tillgänglig litteratur om ämnet HCI, blir läsaren ständigt påmind om att det trots ökade kostnader är viktigt att ta med användarna i designprocessen. Vilka är då de bakomliggande orsakerna? Magnus Lif nämner följande tre punkter i sin uppsats *Adding Usability*:

Demokrati. Användarna har rätt att tycka till vid beslut som rör deras arbetsplats.

Effektivitet, expertis och kvalitet. Effektivitet och kvalitén på mjukvaran förbättras genom att ta med riktiga användare, eftersom de är experter på det egna arbetsområdet.

Engagemang. Om de slutgiltiga användarna får vara med och utveckla systemet, är de mer motiverade att lära och använda det när systemet väl installeras.

Dessutom vet användaren i de flesta fall vad deras system borde innehålla i form av informationsbehov och funktionalitet.

Det har länge stått klart för systemutvecklarna att användarna är en faktor att ta stor hänsyn till, men de har inte riktigt vetat hur detta skall gå till. Det är först på senare år som standards och generella principer har utvecklats för i vilka stadier användaren skall involveras.

För att skapa ett funktionellt gränssnitt behövs dock mer än bara åsikter om vad systemet bör innehålla för funktioner. Det krävs också en gedigen kunskap inom HCI och användare av informationssystem har sällan haft tillfälle eller överhuvudtaget intresse att lära sig detta. Därför kan det vara en bra idé att låta användaren bestämma vad systemet skall innehålla, och sedan låta människor med kunskap inom HCI ta beslut rörande gränssnittets utseende.

Vad väntar oss i framtiden?

När det talas om datorer idag, är det ofta med syftning på en PC med skärm, tangentbord och mus, och denna utrustning kommer nog att finnas kvar i många år framöver. Datorer kommer dock med all säkerhet att spela en allt större roll i vårt framtida vardagliga liv, fast inte på det sätt vi är vana vid. Många av våra vanliga hjälpmedel idag kommer att datoriseras i morgon. Skillnaden är att vi inte kommer att tänka på dem som datorer utan helt enkelt som en integrerad del i vår vardag.

Virtuell verklighet kommer också att ha en stor betydelse i många människors vardagliga liv. Ett framtida scenario kan vara att det inte är nödvändigt att åka till USA för att möta sina släktingar när det istället går att träffa dem virtuellt i vardagsrummet. Datorer kommer att användas av mer människor på jobbet och i hemmet, både för nöjen och för att lösa praktiska problem. Vi har redan bevittnat en otrolig explosionsartad utveckling i användandet av Internet för alla möjliga slags tjänster.

Oavsett vilka tekniska maskiner som än kommer att användas i framtiden, kommer det fortfarande att vara viktigt att förstå användarna och deras behov. Ämnet HCI är alltså något som kommer att följa oss så länge det finns datorer att arbeta med.

ATT DESIGNA ETT ANVÄNDARGRÄNSSNITT

Introduktion

Design är huvudsakligen en fråga om att optimera användargränssnittet baserat på olika krav som ställs på systemet. Kraven kommer från användarna, som är experter på sitt arbetsområde, och vet hur de vill arbeta med det nya datasystemet. Olika grupper av användare har dock olika behov som skall mötas i designen. Det är viktigt att ta hänsyn till deras kunskap, erfarenhet, utbildning och även fysiska attribut. Även den tekniska miljön begränsar utrymmet för design. Storleken på skärmen och resolutionen sätter till exempel begränsningar för hur mycket information som kan visas samtidigt på skärmen. Designerns uppgift är att optimera användargränssnittet baserat på alla dessa förutsättningar, för att skapa den bästa möjliga lösningen för att stödja användarna i deras arbete.

I de allra flesta fall är det systemutvecklaren själv som står för designen av användargränssnittet. Tyvärr. För ofta anser denne att ”om jag bygger ett system där funktionerna fungerar som de ska, så måste ju användarna vara nöjda!”. Fel. En systemutvecklare är inte en ”typisk” användare när det gäller datasystem. En systemutvecklare är ingen expert på arbetet som användarna skall använda systemet till. En systemutvecklare känner inte till miljön där systemet skall användas. Alla dessa faktorer är viktiga vid design av användargränssnitt. Därför bör någon med insikt och kunskap inom dessa områden ta hand om denna del i utvecklingsarbetet, och därför bör användaren ha en stor roll i processen att ta fram ett gränssnitt.

På senare år har grafiska användargränssnitt blivit vanliga, och det är nu möjligt att använda färger, fonter och 3-D representationer i applikationerna. Genom att använda grafik har det blivit möjligt att skapa informationsrika, unika och effektiva gränssnitt som är lätta att använda och lära sig. Å andra sidan har det också ökat risken att ta fel beslut vid design. Felaktig användning av grafiska verktyg kan faktiskt resultera i användargränssnitt som är mindre effektiva än det gamla alfanumeriska gränssnittet. Utvecklingen har gjort att det nu ställs högre krav på designern än tidigare.

Syfte & metod

I denna del vill vi på ett konkret sätt beskriva hur ett användargränssnitt på bästa sätt bör utformas. Vi börjar med att granska användarna lite mer ingående. Detta för att verkligen poängtera att det är viktigt med användarmedverkan vid gränssnittsdesign och orsakerna för detta. Sedan beskrivs mycket kort det grafiska användargränssnittets historia och dess för- och nackdelar, innan vi kommer in på kapitlet om de grundläggande principerna vid design av användargränssnitt. De allmänna riktlinjerna skall inte ses som några konkreta

förmaningar om hur utseendet på gränssnittet *skall* vara, utan mer som goda råd som bör övervägas av designern. Layout-avsnittet behandlar fönster, menyer och olika former av kontroller som ett system kan använda sig av. Syftet med detta är att ge en klar bild av när, var och hur dessa bör användas för maximal effekt.

Materialet till denna del har vi samlat genom litteratur och artiklar både från tidningar och på nätet.

Användarna

För att kunna designa ett informationssystem som stöder användarna i deras arbete, är det nödvändigt att förstå dem och deras behov. Även om många olika typer av människor slutligen kommer att utnyttja systemet, måste designern fokusera på en eller några användare vilka kan fungera som målgrupp, och kring vilkas behov systemet designas. Generellt sätt är det så att ju mer målinriktat systemet är, ju mer framgångsrikt blir det. Frestelsen med att tänka i banorna "Mitt system är till för alla!", bör undvikas. Designern får välja sin målgrupp och utforma gränssnittet med dem hela tiden i åtanke.

När målgruppen väl har utsetts, är nästa steg att ta fram en användarprofil. Detta är en samling fakta om användarna, som ger svar på följande frågor:

- Vad vet användarna om deras arbetsområde?
- Vilka termer och begrepp i detta arbetsområde känner användarna till?
- Vad vet användarna om datorer?
- Vilken miljö arbetar de i?
- Vilka jobb har de?
- Finns det särskilda psykiska, sociala, kulturella eller andra faktorer att ta hänsyn till vid design för dessa användare?

Ju mer kunskap det finns om användarna, desto lättare är det att fatta beslut vid designen. Speciellt då situationen är sådan att någon funktion måste väljas bort till förmån för någon annan.

Låt oss tänka på saken!

Kognitiv psykologi, dvs att förstå hur människor tänker och uppfattar sin omgivning, är viktigt för designers att ha kunskap om när de vill lära känna användaren. Många modeller har tagits fram för att beskriva vad som sker i den mänskliga hjärnan vid utförandet av en uppgift, och några av dessa har varit till stor hjälp för forskare inom HCI.

En av de mest erkända modellerna av mänskligt minne är **Stage Theory**, skriven av Jean Piaget (1969). Enligt denna teori, lagrar alla människor information dels i ett **korttidsminne** och dels i ett **långtidsminne**. Det

mänskliga korttidsminnet kan endast behandla en begränsad mängd med information, och är även begränsat vad gäller tid. Lagrat material försvinner efter ungefär 15 sekunder om det inte processas vidare, till exempel genom upprepning. I motsats till korttidsminnet är långtidsminnets kapacitet enorm, både vad gäller informationsmängd och i tidspann. Information i detta minne kan nås via utlösare såsom ord, lukt eller ljud.

Hur minnet fungerar har en stor betydelse vid arbete med datorer. På grund av den begränsade kapaciteten av korttidsminnet, bör all information som behövs för att fatta ett beslut vara synlig samtidigt på skärmen. Annars måste användaren lagra information i långtidsminnet eller göra skriftliga anteckningar, vilket gör arbetet långsammare.

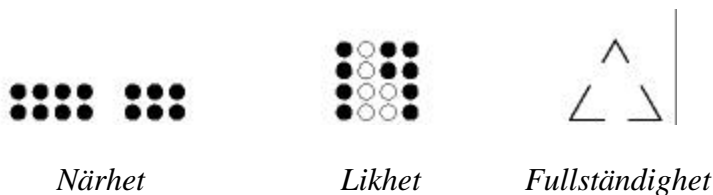
Kognitiva psykologer studerar även mönsterigenkänning. Detta är viktigt när beslut fattas som har att göra med användning av färger, fonter, storlekar och hur information ska grupperas på skärmen för att optimera söknings- och läsningsprocessen.

Det har funnits flera teorier under åren om hur mönster skall klassificeras. En av dessa är **The Gestalt approach**, framtagen av bland annat Erving och Miriam Polster, i vilken dessa tre "lagar" finns att tillgå:

Närhet. Ju närmare två figurer är varandra, ju troligare är det att dessa grupperas tillsammans.

Likhet. Objekt som ser likadana ut grupperas vanligen tillsammans.

Fullständighet. Ofullständiga figurer ses oftast som fullständiga.



Figur 1. Exempel på lagar i "The Gestalt approach"

Erfarna vs ovana användare

Kunskapen om datorer och datasystem bland slutanvändarna kan variera stort. Detta är varken ovanligt eller särskilt förvånande, och något som en designer måste ta hänsyn till. Användare med erfarenhet av datorer kan arbeta med datasystem på ett annat sätt, eftersom de vet hur datorn vanligtvis reagerar vid användning. De som arbetat mycket med ett visst system kan utnyttja genvägar via tangentbordet, medan den ovane användaren vanligtvis använder sig av musen.

De flesta datasystem bör designas för att passa både ovana och erfarna användare, dvs kombinera effektiviteten i det dagliga arbetet med att systemet skall vara lätt att förstå. Det finns många exempel på riktlinjer för erfarna användare som står i konflikt med riktlinjer för ovana användare. Det är till exempel viktigt för den ovane att mängden av information på bildskärmen hålls nere till ett minimum, medan däremot den erfarna användaren kräver, och klarar av, mycket mer information.

Det är också viktigt att ha i åtanke att det för en komplex applikation kanske bara finns några få som är experter i alla aspekter av systemanvändningen. De flesta blir med tiden duktiga på de funktioner som de använder ofta, men utnyttjar sällan eller kanske aldrig övriga funktioner. Det ligger ofta en fara i att tro att alla användare är experter och enbart designa för dessa.

Det är inte bara insidan som räknas...

Inom HCI har stor möda lagts ner på att förbättra effektiviteten i systemen. Men ett system skall inte enbart vara funktionellt. När en kund väljer mellan två produkter, jämför han inte bara användbarheten av produkterna, det måste också vara behagligt och estetiskt. En produkt skall via sitt yttre kunna förmedla till användaren vilken slags produkt det är och vad som kan förväntas av den. Hur många människor vill till exempel att snabba bilar ser ut som traktorer? Detta gäller också vid design av användargränssnitt. Det skall vara möjligt att se om systemet är tänkt att användas i en bank eller om det är utvecklat för läkare på ett sjukhus. Utseendet och "känslan" på ett system kan vara avgörande när det gäller att få användaren att känna sig väl till mods med systemet. Det blir också ett sätt att tävla med rivaliserande företag på marknaden. Ett informationssystem måste vara tillfredsställande, behagligt, vackert och spännande för att bli en populär produkt!

Det grafiska användargränssnittet har erövrat världen

På 70-talet regerade det teckenbaserade gränssnittet i världen med timesharing och kommandorader i spetsen. Men på Xerox Palo Alto Research Center (PARC) i Kalifornien tyckte de att det var dags för förändring. De började ta fram ett grafiskt användarsnitt, som sedan vidareutvecklades av Apple för Macintosh, Microsoft för Windows och Motif för UNIX Workstations till de användargränssnitt som vi använder idag med fönster, ikoner, menyer och pekverktyg. Dessa gränssnitt har kommit att kallas **GUI:s** efter den engelska termen *Graphical User Interface*.

Det viktigaste med grafiska användargränssnitt kan kort beskrivas med att det har tagits fram för att förbättra arbetssituationen för användaren. Gränssnittet skall kunna kommunicera visuellt och uppmuntra till interaktion med systemet.

För att underlätta detta används komponenter som komplementerar en användares erfarenheter från verkligheten, och som gör att denne förstår och accepterar gränssnittet lättare.

Fördelar med GUI:s är att de är lätta att lära, lätta att använda, och det är enkelt att ta med sig kunskap från den ena applikationen till den andra, på grund av likheter i utseende och funktionalitet. De är ett naturligt val av gränssnitt just därför att de är användarvänliga, färgrika, fönsterbaserade gränssnitt som förlitar sig på ikoner och ett språk som alla förstår. Detta har också lett till att andra grupper av användare, som till exempel barn som ännu inte lärt sig läsa eller skriva och icke professionella hemanvändare, har kunnat börja använda datorer.

Men det är inte bara fördelar med GUI. Många applikationer är så stora att användarna praktiskt taget drunknar i funktionaliteten och inte vill uppgradera till nya versioner, utan föredrar att hålla sig till den lilla mängd de kan. Många användare spenderar alltför mycket tid med att manipulera gränssnittet, inte applikationen. När det bara fanns radorienterade stordatorgränssnitt med ett visst antal positioner per rad kunde fel inte åstadkommas på så många sätt som är möjligt idag.

Direkt manipulation

Termen ***direkt manipulation*** myntades först av Ben Shneiderman för att beskriva nya gränssnitt med följande egenskaper:

- Objekten finns hela tiden tillgängliga på skärmen.
- Fysiska händelser eller knappar med etikett används istället för kommandospråk.
- Snabbt tillgängliga återställningsfunktioner, vilka ger direkt synligt resultat.

Generellt sett kan användare i ett gränssnitt med direkt manipulation utföra uppgifter direkt på synliga objekt, så som att ersätta, flytta och ta bort information. Ibland kallas dessa användargränssnitt ***”peka och välj”-gränssnitt*** och inkluderar ofta något slags pekanordning, som exempelvis *mus*, *trackball* eller *touch screen*. De har ofta mycket grafik för att beskriva objekt och händelser.

Tricket att skapa ett system med direkt manipulation är att hitta en bra modell med metaforer som representerar verkligheten på ett lämpligt sätt. Många system använder ofta skrivbordet som metafor, där ikoner av välkända kontorsföremål syns på skärmen, till exempel kalendrar, mappar och dokument.

Ett av de gränssnitt med direkt manipulation som är enklast att förstå är det ***formulär-baserade gränssnittet***. Användaren presenteras för det aktuella formuläret där fälten har etiketter som talar om vilken information som ska

fyllas i. När den efterfrågade informationen är ifylld fortsätter användaren till nästa fält, vanligtvis genom att använda tabb-knappen eller musen. Det kan också finnas funktioner som kontrollerar att rätt information fyllts i.

Några **fördelar** med direkt manipulation:

- Användarna känner att de har **kontroll** över datorn.
- Användarnas **inlärningstid** är relativt kort.
- Användarna får **direkt feedback** på sina handlingar. Fel upptäcks och kan snabbt rättas till.

Problem med att designa gränssnitt för direkt manipulation:

- Hur väljs rätt metaforer? I vissa fall är detta ganska självklart. Som ett exempel Apples Macintosh ta bort-funktion som går ut på att dra ikonerna till papperskorgen, vilket är en uppenbar metafor för att kasta bort någonting. Det är dock inte alltid lätt att hitta en passande modell om det inte finns någon klar anknytning till verkligheten som kan användas.
- Om användarna har ett stort informationsutrymme, hur kan de då navigera runt och alltid vara medvetna om sin nuvarande position? Detta problem är jämförbart med det problem som uppstår vid användning av menyer med flera nivåer.

Gränssnitt med direkt manipulation är något komplexa och kräver mycket av underliggande processorer och minne. Detta spelar dock mindre roll i dagens läge, eftersom de processorer som nu tillverkas är otroligt snabba och inte speciellt dyra.

Grundläggande principer vid design av användargränssnitt

”När det bara fanns radorienterade stordatorgränssnitt med ett visst antal positioner per rad kunde man inte göra fel på så många sätt ... Med de nya verktygen däremot öppnar sig fantastiska möjligheter att designa omöjliga gränssnitt. Man kan bygga knappar och generera fönster och länka vidare och skapa en faslig oreda...”

Ur tidningen Datateknik nr 29, 1998

Att optimera användargränssnitt är ingen lätt uppgift och att beskriva exakt hur det görs är inte möjligt, eftersom det inte finns några generella lösningar. Vad som är optimalt i det ena fallet kan vara helt galet i ett annat. Trots detta har det utvecklats en hel del standards och riktlinjer vilka kan fungera som en checklista för designers vid framställandet av ett användarvänligt, funktionellt och effektivt användargränssnitt.

Allmänna riktlinjer

De designprinciper som presenteras nedan är en sammanställning av ett antal författares teorier¹ och är grundläggande för design av effektiva gränssnitt, vare sig det gäller traditionella GUI-miljöer eller webbsidor. De har varit aktuella ända sedan användargränssnitt för ett antal år sedan började användas, och det har visat sig att de gäller oberoende plattform. Orsaken till detta är att de flesta principer vänder sig till människors förmåga och psykologiska natur, snarare än de olika plattformarnas konventioner.

Lär känna användaren. Om en designer av ett användargränssnitt tror att alla användare tycker och tänker likadant, är denne riktigt illa ute. Till exempel så skiljer sig en erfaren användares krav på ett system från den oerfarnes. Det är viktigt att lära känna och förstå de enskilda individerna i användargruppen samt detaljerna i deras arbete. Designers bör också kunna grundläggande kognitiv psykologi, dvs en förståelse av de generella styrkor och svagheter hos det mänskliga sinnet.

Förutse användarens behov. Applikationer bör försöka förutse användarens behov och önsningar. Förvänta dig inte att användaren själv söker eller samlar information eller tar fram nödvändiga verktyg. Förse användaren med all nödvändig information samt verktyg till stöd för varje steg i processen.

Visa systemstatus. Det är viktigt att användaren kan orientera sig i systemet, dvs att han vet var i systemet han befinner sig, hur han kom dit och hur han kan fortsätta. Detta kan uppnås genom att via en översikt presentera den väg som tagits genom systemet. Det måste också stå klart om det finns ytterligare information som inte visas och hur den går att få tag i.

Ge direkt respons. Datorn måste svara direkt på input, annars kan användaren tro att systemet lagt av och frenetiskt börja trycka på knappar. En undersökning² visade att när datorn inte visuellt svarade på en knapptryckning, tog det bara 8.5 sekunder för hälften av deltagarna att tro att maskinen hade hängt sig och tryckte på RESET-knappen. Detta bör och kan undvikas, till exempel med timglas eller meddelanderutor som informerar om vad som sker och hur lång tid det beräknas ta. Dessa ska vara placerade på ett sådant sätt att användaren ej ska behöva leta reda på informationen. Det skall dock inte överdrivas med stora blinkande ikoner som tar koncentrationen ifrån uppgiften.

Framhäv viktig information. Sidor bör *inte* innehålla information som är irrelevant eller som sällan används. All extra informationsmängd drar uppmärksamheten från de relevanta uppgifterna och gör att användaren blir distraherad och mindre effektiv. Lägg istället information som sällan används så att det går att nå den med länkar. Då finns detaljerna där när de behövs,

¹ Se referenslista a)

² Exemplet är hämtat ur Adding Usability av Magnus Lif, 1998

samtidigt som de inte inkräktar på den viktigare informationen. Se också till att användaren lätt kan nå den generella informationen även nere på detaljnivå.

Objekt, händelser och val som används frekvent skall dock placeras väl synliga. Viktig information, såsom instruktioner för hur systemet används, bör ligga så att de enkelt kan nås när det behövs.

Det är bättre att förse gränssnittet med mindre funktionalitet som kommer att användas, än så mycket funktionalitet att gränssnittet blir hopplöst komplext.

Håll teknologin osynlig. Vad som egentligen händer inuti datorn skall vara osynligt för användaren, eller i alla fall presenteras på ett enkelt och familjärt sätt så att användaren inte konfronteras med obekanta termer och processer som denne inte förstår.

Tillåt flexibilitet och effektivitet i användningen. Funktioner som kan snabba upp användningen för den vana användaren ses ofta inte av den ovane användaren, och tillåter på så sätt systemet att hantera båda parter. Samtidigt tillåter det också användaren att skraddarsy funktioner som denne använder ofta, vilket ger mer användarkontroll.

Utnyttja direkt manipulation. Ett bra grafiskt gränssnitt tillåter användaren att känna kontroll. Istället för att komma ihåg en massa kommandon och parametrar, kan användarna lära sig några få enkla gränssnittsobjekt såsom mappar eller knappar, och sedan använda sig av denna kunskap i olika situationer. Användaren utför direkt uppgifter på synliga objekt genom att använda sig av musen, och får direkt se resultatet. Gränssnittet blir mer direkt och lättare att lära och använda.

Använd metaforer från verkligheten. Att använda sig av termer och objekt som är kända för användaren sedan tidigare, är ett bra tillvägagångssätt att tillverka ett gränssnitt som är lätt att förstå för nya användare. Idén är att eftersom användare redan vet hur dessa verktyg används i verkligheten, så kan de nu ana sig till hur de används i systemet, när de ser dem på skärmen. Några klassiska exempel är mappar för filer, penslar och verktyg i ritprogram, postlådor i mail-program, och naturligtvis skräpkorgen för sådant som ska slängas bort.

Använd standardvärden. Idén är att designa i nivåer så att den nye användaren inte behöver lära sig allt vid första mötet, utan kan lära sig grunderna och sedan gradvis lägga till mer avancerade och sofistikerade funktioner till deras repertoar i egen takt. Ett sätt att göra detta är att använda sig av standardvärden.

Var konsekvent. Systemet bör vara konsekvent i funktionalitet och layout. Funktioner som fungerar på ett sätt i en del av systemet bör också fungera på liknande sätt i övriga delar. Något som även gäller mellan de olika systemen. Har användaren lärt sig något en gång, kan denne använda sig av denna kunskap om och om igen. Detta minimerar inlärningstiden hos användaren, och förbättrar prestationen samt den personliga tillfredsställelsen att verkligen kunna något.

Men lika viktigt som det är att vara visuellt konsekvent när objekt uppför sig på samma sätt, lika viktigt är det att vara visuellt inkonsekvent när objekt uppför sig olika. Objekt skall användas i samklang med vad de utför. De som uppför sig olika skall ha olika utseenden.

WYSIWYG – What You See Is What You Get. Dokument på skärmen skall se ut precis som de gör när de skrivs ut. Fet text på skärmen skall alltså vara fet text vid utskrift.

Låt systemet matcha verkligheten. Låt systemet tala användarens språk. Använd ord, fraser och koncept som är välkända för användaren - undvik systemorienterade termer. Följs dessa konventioner verkar informationen mer logisk och naturlig för användaren.

Var förlåtande. Användare gör fel. Bra program hanterar detta genom att låta dem ångra senaste handlingen, eller återgå till senaste sparade versionen. Om användare är på gång att utföra någon eventuellt skadlig handling som inte går att ångra, skall datorn informera om detta och fråga om de vill fortsätta. Felmeddelanden bör uttryckas på ett enkelt sätt - använd inte koder. Tala om vad det exakta felet är, indikera problem och föreslå en lösning på ett konstruktivt sätt.

Om systemet ej är feltolerant blir användaren försiktig och vågar inte utforska nya omgivningar. Kom ihåg att det enda som är bättre än ett bra felmeddelande är en noggrann design som förebygger att fel uppstår över huvudtaget .

Förse användaren med hjälp och dokumentation. Hjälp ska ske på användarens initiativ. Denna hjälp skall vara enkel att nå och förstå. Om ingen annan lösning är möjlig, ge hjälp på initiativ av systemet.

Även om det är bättre ifall ett system kan användas utan dokumentation, kan det ibland bli nödvändigt att förse användaren med viss information. Denna ska vara lätt att hitta, den ska fokusera på användarens uppgift, lista konkreta steg att utföra och inte vara för lång.

Designa så att det är lätt att lära och lätt att använda. System skall vara lätta att lära för novisen och effektiva och enkla att använda för experten. Många av föregående principer leder till detta, och det är vad användargränssnittsdesign går ut på.

Det är svårt att kombinera alla ovanstående punkter. Om det läggs större vikt vid en funktion, kan designern komma att behöva ge efter på något annat. Ofta är en kompromiss nödvändig. Tänk på målgruppen!

Layout

Eftersom det grafiska gränssnittet har fått ett så starkt fotfäste i datorvärlden, kan en designer av användargränssnitt inte längre undvika frågor som berör positionering och disponering av fönster, menyer och kontroller. Här behandlas de allra vanligaste komponenterna, och det ges tips och råd vid användning av dem.³

Fönster

Vid design av fönster, bör stor vikt läggas vid storlek, position, form och stil. Storlek och form beror till stor del på vilka fält som skall finnas med, men användningen av flera fönster och flikar i ett fönster ger designern ganska mycket frihet.

Storlek och form. Det finns inga konkreta bevis för att en viss höjd eller bredd på ett fönster skulle vara det bästa. Så länge alltför breda eller höga fönster undviks, så har designern frihet att använda vilken form och storlek som bäst passar för placeringen av fält.

När möjligheten finns bör fönstret designas så att alla kontroller är synliga utan att behöva scrolla. Om detta inte är möjligt bör de viktigaste kontrollerna ligga synliga. Kontroller som svarar på en dialog, exempelvis OK-knappar bör alltid vara lättillgängliga. Fönster som är justerbara bör behålla den storlek som användaren väljer. Om denne stänger fönstret och sedan öppnar det igen skall det fortfarande ha den senaste valda storleken.

Konsistens bör vara en viktig faktor. Designern kan förstärka relationerna mellan likartade fönster genom att konsekvent använda samma storlek och form.

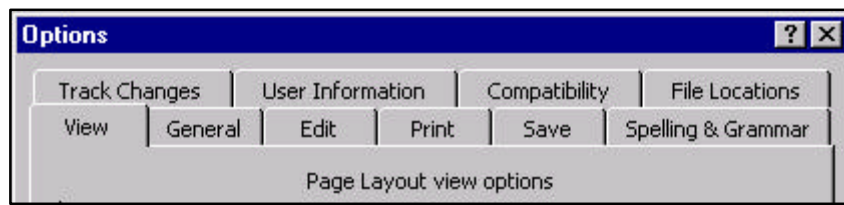
Position. Skärmens yta är en begränsad resurs. När mycket information skall visas samtidigt är det viktigt att noga överväga hur den ska disponeras. Det ska vara klart vilka huvudenheter på skärmen är, och hur dessa är relaterade till varandra.

³ Se referenslista b)

När ett fönster eller en dialog först skapas, måste designern placera det på det mest lämpliga stället för användaren. Två olika riktlinjer bör övervägas:

- Lagg fönstret nära platsen där det aktiveras. Att tvinga användaren att föra musen tvärs över skärmen från den punkt dialogrutan aktiveras till den plats där dialogrutan befinner sig, leder snabbt till irritation.
- Täck inte fält som användaren behöver se. Användaren blir frustrerad om han tvingas flytta på fönstret så snabbt han öppnat det på grund av att det ligger i vägen för den information han behöver se i det tidigare fönstret.

Fönster och flikar. Vid användning av multipla fönster är det viktigt att designern behåller stabiliteten i dem. Om fönstren ändrar storlek och form för varje nytt fönster som presenteras, förlorar användaren känslan av integritet som var meningen från början med de multipla fönstren. Det kommer istället att verka som ett antal orelaterade fönster ligger på varandra istället för en integrerad helhet.



Figur 2. Exempel på flikar i Microsoft Word.

Om flikar används i stället fås automatiskt visuell stabilitet, eftersom varje flik har samma storlek och form. Andra fördelar med flikar är:

- Alla grupperingar med val är synliga på samma gång.
- Användaren kan nå alla olika grupperingar med ett klick.
- Metaforen av mappar med flikar är fysiskt konkret och välkänd även för en nybörjare.
- Jämfört med listor och fält med ikoner så tar flikar upp relativt lite skärmyta.

Dessa fördelar gör att flikar är ett naturligt val för att dela upp flersidiga dialogrutor och andra grupper av relaterad information. I användartestning⁴ har det framkommit att flikar fungerar något bättre än popup-listor vad det gäller att hjälpa användaren att hitta olika alternativ av val.

En nackdel med flikar är att de är begränsade i antal beroende på hur bred titeln är. Det går till viss del att komma runt detta problem genom att använda sig av förkortningar, men det bör också hållas i åtanke att om applikationen ska köras

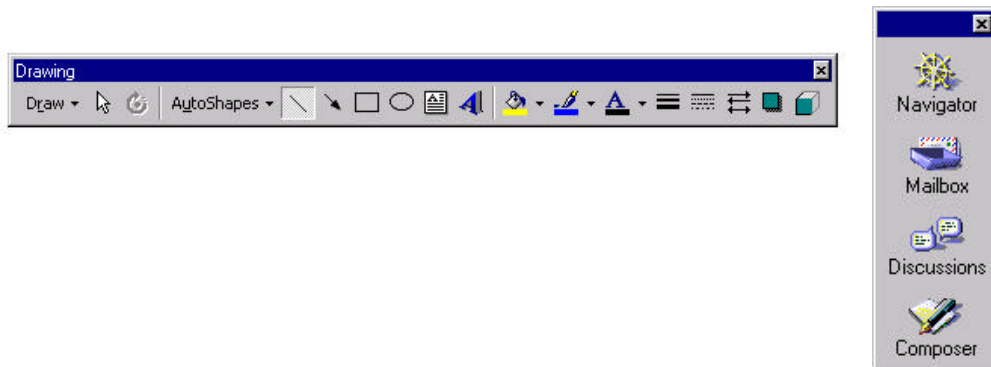
⁴ Exemplet är taget ur Practical user interface design, Carl Zetie

på olika språk, så varierar bredden beroende på vilket språk som används. Det är också viktigt med noggrannhet i användningen av förkortningar, dessa måste vara lätta att förstå och logiska för användaren.

Verktygsfält & paletter

För att underlätta användningen har de flesta moderna applikationer någon form av verktygsfält eller paletter. I vissa fall så många att om alla är aktiverade tar de mer plats än det egentliga dokumentet.

Verktygsfält och paletter har egentligen samma syfte, den stora skillnaden är att verktygsfält är fästa i ramen på fönstret och kan antingen vara vertikala eller horisontella. En palett är vanligtvis ett självständigt flytande fönster.



Figur 3. Exempel på paletter. Ovan från Microsoft Word, och till höger från Netscape Navigator.

Syftet med verktygsfält är att ge tillgång till vanliga händelser såsom *Öppna fil*, *Kopiera* och *Skriv ut*, genom bara ett klick. Därför består ett verktygsfält oftast av knappar, och för att få plats med så många funktioner som möjligt är dessa knappar ganska små och försedda med ikoner, som talar om vilken funktion knappen har. Verktygsfältet kan också innehålla checkboxar med ikoner, dessa ser ut som knappar men när användaren klickar på dem så förblir de nedtryckta tills dess att de åter aktiveras. Vanliga exempel på när dessa används är när det önskas fet stil och understrykningar. På samma sätt kan det också finnas radioknapps-funktioner försedda med ikoner, där bara ett alternativ kan väljas åt gången. Så fungerar till exempel funktionen för textuppställning där *centrerad*, *vänsterställd* och *högerställd* text kan väljas. Verktygsfält har tre viktiga användningsområden:

- De gör **globala, ofta använda funktioner, snabbt tillgängliga**. Verktygsfält är ett utmärkt sätt att uppfylla målen om minsta möjliga arbete för användaren. Att skriva ut kan till exempel göras med en knapptryckning och utskriften sker då med standardinställning. Först när användaren behöver andra inställningar, krävs mer arbete.

- De gör **viktiga funktioner lätta att hitta**. I en komplex applikation, kan antalet menyval vara överväldigande stort. Ett sätt att lösa detta är att anpassa verktygsfältet beroende på dokument eller applikation. Detta används mycket i bland annat mail-funktioner där verktygsfältet ändras beroende på om du ska skicka eller läsa mail.
- De kan vara **visuellt attraktiva**. Förutsatt att den grafiska designen är välgjord kan ikoner och annat i verktygsfältet förse en annars ganska tråkig applikation med färg, vilket attraherar användaren mer. Detta är dock underordnat de två ovanstående skälen.

Menyer

Navigering i informationssystem tar mycket tid. Detta kan underlättas genom att organisera stora mängder av kommandon och händelser i hierarkiska menyer, som vanligtvis ligger i överkanten på applikationens huvudfönster.

Meny-objekt kan innehålla undermenyer, även kallade **pop-up-, pull-down- eller kaskadmenyer**. I teorin finns möjlighet till hur många nivåer som helst på undermenyerna, men av praktiska skäl bör fler än tre nivåer undvikas, eftersom det annars blir svårt för användaren att följa meny-strukturen och hitta rätt kommando.

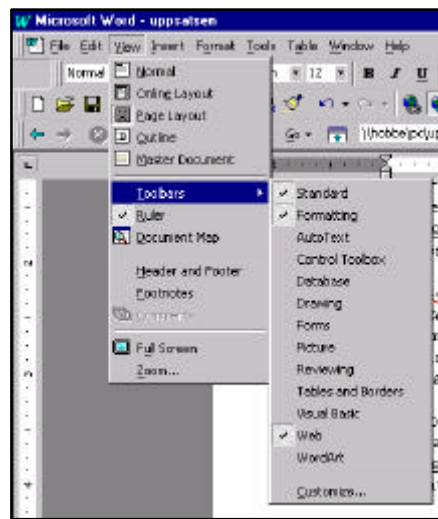
Eftersom menyer alltid är tillgängliga i en applikation bör generella kommandon och händelser läggas där. Sådant som är specifikt för vissa verktyg bör läggas direkt i dialogrutor. Menyer kan också förändras dynamiskt så att vissa av kommandona är inaktiverade när de ej är aktuella att användas.

Det finns flera teorier om hur menyer byggs upp efter **substantiv/verb-modellen**, till exempel *Arkiv/Spara*, för att reflektera objekt/händelse-idén som karaktäriserar GUI-applikationer. I praktiken däremot hittas ofta andra varianter, som till exempel verb/verb (*Redigera/Radera*) och verb/substantiv (*Visa/Verktysfält*).

I verkligheten bryr sig inte användaren alls eller väldigt lite om dessa skillnader, utan det som är viktigt är att kategorierna är väl organiserade, har lämpliga och förklarande namn, och att objekten i menyn lätt kan skiljas åt och kännas igen. **Enkelhet är viktigt.**

Det finns flera olika teorier om i vilken ordning meny-objekt ska arrangeras: **alfabetiskt, mest använda, grupperade beroende på funktion** och så vidare. I verkligheten tycks detta spela mindre roll så länge ordningen är logisk. Det vanligaste är att gruppering sker beroende på funktion och därefter mest använda objekt. Har en mjukvarukonstruktör ett flertal liknande applikationer ute på marknaden är det viktigt att vara konsekvent och inte ändra för mycket i

menyerna bland de olika applikationerna, eftersom användaren lär sig var funktionerna finns och tar för givet att de ska ligga på samma plats.



Figur 4. Exempel på meny med två nivåer, som visar undermeny, inaktiverade kommandon och uppbyggnad med verb/substantiv.

Menyer och val. På samma sätt som ett stort antal kommandon kan presenteras på en liten yta, kan menyer också användas till att låta användaren välja mellan olika möjligheter. Ett vanligt sätt är att låta användaren markera sitt alternativ med en bock för att visa att det är aktiverat eller valt.

Alternativ kan också representeras genom att ändra ord i det, exempelvis *linje dold* ändras till *linje visas*, istället för att använda en bock. Detta är dock inte att rekommendera eftersom det kan bli väldigt förvirrande för användaren; när det står *linje dold*, betyder det att linjen är dold eller att den blir det när användaren klickar på alternativet?

Popup-menyer. I de flesta applikationer finns ytterligare en variant av menyer tillgängliga, vanligtvis kallade popup-menyer. Sådana menyer aktiveras oftast genom att välja ett objekt och sedan klicka på den högra musknappen. De vanligaste händelserna för objektet blir då tillgängliga.

Popup-menyer är mycket effektiva vad det gäller att lägga funktionaliteten nära muspekarens position. Där bör det inte placeras några onödiga kommandon som kanske är lika lättillgängliga med kortkommandon.

Det bör också noga övervägas vilka kommandon som ska placeras högst upp i popup-menyerna, eftersom risken för att användaren av misstag råkar klicka på dessa är ganska stor. Lämpliga kommandon att placera där är sådana som inte kan ställa till med någon större skada, exempelvis *Hjälp* och *Kopiera*.



Figur 5. Exempel på popup-meny, Microsoft Word.

Ett problem som ofta uppstår vid användning av popup-menyer är hur användaren ska kunna veta när menyn finns tillgänglig och hur den tas fram. Ett av det vanligaste sätten att plocka fram menyn är att använda högerknappen på musen, men det finns flera sätt. Om popup-menyer används frekvent kan markörens form ändras när dessa menyer finns tillgängliga för att göra användaren medveten om att de finns.

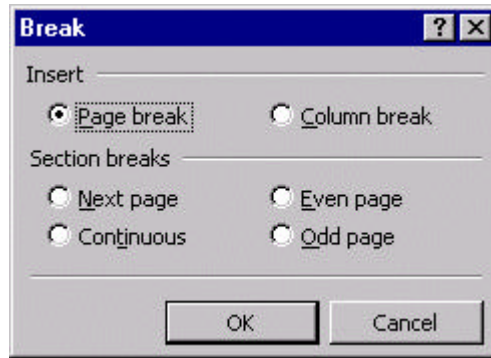
Input med mus tar normalt längre tid än input från tangentbordet. Därför är det viktigt att förse användaren med möjligheten att använda sig av genvägar och standardval, till exempel Microsofts *Ctrl+C* för att kopiera. Använd koder i menyn för att visa om det är möjligt att använda ett specifikt alternativ eller inte.

Kontroller

Till kontroller räknas de komponenter på bildskärmen som användaren kan interagera med, till exempel **knappar** och **textfält**. De flesta kontroller används både för input och output där användaren kan använda dem antingen för att ta fram befintliga värden eller för att lägga till nya, men givetvis finns det också de kontroller som används enbart i ena fallet. Nedan följer några riktlinjer vid användning av de allra vanligaste kontrollerna.

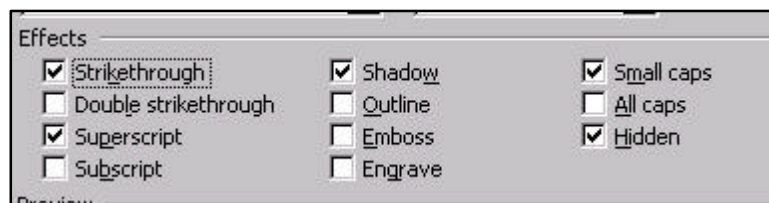
Radioknappar. När en användare ska välja mellan ett antal fasta värden passar grupperade radioknappar alldeles utmärkt, där enbart ett värde kan vara aktiverat åt gången. Observera att radioknappar passar bäst när antalet värden är få, eftersom en större mängd får skärmen att se rörig ut, vilket i sin tur leder till att det blir svårt att läsa informationen. I de fall där det förekommer ett större antal värden bör istället exempelvis listor övervägas.

Grupper med radioknappar kan läggas ut vertikalt - den ena ovanför den andra, horisontalt - tvärs över fönstret, eller både och. Vilken layout som än väljs är det viktigt att den stödjer knapparnas inbördes relation, dvs hur väl de hör ihop med varandra. Detta kan till exempel göras genom att rama in dem.



Figur 6. Exempel på radioknappar i Microsoft Word.

Check-boxar. När ett värde kan anta enbart booleska värden, som till exempel *ja/nej* och *på/av* är det lämpligt att använda sig av en checkbox. Beroende på plattform kan en ifylld box presenteras med ett kryss eller en bock. Vid användning av bockar uppstår det sällan problem, medan däremot kryss kan ha en dubbeltydig mening. För vissa användare och kulturer betyder nämligen kryss *fel* och används för att utesluta någonting.



Figur 7. Exempel på checkboxar med bockar taget från Microsoft Word.

Checkboxar kan alltså innebära problem. Användaren kan ha svårighet med att skilja markerade och omarkerade boxars betydelse. Tyvärr finns det inte mycket att göra åt detta, förutom att försöka få användaren uppmärksam på vad markerad och ommarkerad box står för.

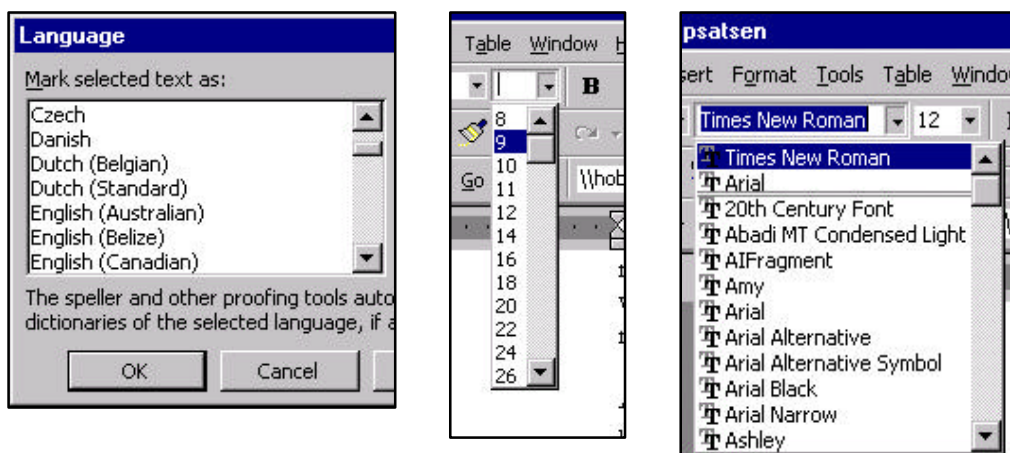
Checkboxar bör alltid få en positiv etikett - om svaret på frågan som ställs blir *ja* skall rutan markeras. Detta för att undvika dubbelt negativa fraser vilket lätt förvirrar.

Ett annat vanligt fel som uppkommer med checkboxar är att det oftast antas att dessa bör användas så fort det är ett val mellan två värden, när de istället bör användas enbart när det är frågan om ett booleskt val. Vid andra tillfällen bör exempelvis radioknappar övervägas.

Combo-boxar och drop-down listor. Olika sorter av drop-down listor och combo-boxar är användbara när dynamiska val ska visas till skillnad från radioknappar som representerar fasta värden. Eftersom en drop-down lista kan visa delar av alla inlästa värden med hjälp av en scrollbar, är de också lämpliga när valen är så pass många att det skulle bli för rörigt att använda radioknappar.

Att ersätta enkla textrutor där inmatning måste ske, med listor av värden underlättar för användaren, eftersom det är lättare att känna igen ett värde än att komma ihåg det, speciellt vid de tillfällen då det krävs exakta värden. Exempel på detta är kundnamn eller företagsnamn där stavningen kan vara svår att komma ihåg.

Det finns tre olika sorters combo-boxar; den enkla där listan alltid är synlig så att användaren kan se vilka val som finns, drop-down där listan visas när användaren väljer att den ska visas och användaren kan också välja att skriva in ett eget värde om så önskas, och sist drop-down där användaren är tvungen att använda ett värde som finns i listan.



Figur 8. Exempel på de olika combo-boxarna, Microsoft Word

Listboxar. Liksom combo-boxar tillåter även listboxar att användaren väljer ett av flera värden i en lista. Skillnaden är att en listbox kan bestå av flera kolumner och på så sätt kan flera värden visas för en post. Detta är mycket användbart för bland annat fakturor och transaktioner.

Knappar. Knappar används för ett ganska litet antal kommandon eller händelser. De bör användas för händelser som är bundna till en specifik dialog, till exempel *OK*- och *Avbryt*-knappar. För mer generella händelser bör istället menyer användas.

Etiketten på knapparna bör vara ett verb som beskriver vilken funktion knappen har, till exempel *Spara* eller ett substantiv som beskriver resultatet av händelsen, exempelvis *Egenskaper*. Det är också vanligt med en ikon på knappen istället för ett ord. Detta måste dock användas med stor försiktighet eftersom bilder ofta ger fler tolkningsmöjligheter än ett ord.

Pilar är ett vanligt förekommande inslag, men måste hanteras med försiktighet. De som pekar åt vänster och höger skapar sällan problem: vänster betyder alltid "föregående" eller "minska" och höger betyder alltid "nästa" eller "öka",

beroende på sammanhang. Pilar som pekar uppåt och neråt å andra sidan kan ibland leda till tvetydighet. En pil som pekar uppåt kan betyda "föregående", men den kan också betyda "öka", liksom en pil som pekar neråt betyder "nästa", men också kan betyda "minska". Gränssnitts-designern bör därför undvika upp och ner pilar när det finns risk för missuppfattningar hos användaren.

Ikoner. Ikoner används på flera olika sätt i en GUI-applikation. En ikon kan till exempel representera en applikation på skrivbordet eller i fönsterhanteraren, där den tillsammans med titel representerar applikationen som ett minimerat fönster. Att titeln står med är viktigt för att användaren direkt ska kunna skilja olika applikationer åt utan att först behöva öppna dem.



Figur 9. Exempel på minimerade fönster, Windows NT.

Kontroller som används mycket är lämpliga att lägga upp som ikoner så att användaren har dem direkt tillgängliga. Det är då viktigt att välja en bild som talar om vad kontrollen gör. Exempel på kontroller som ofta har ikoner är *Skriv ut* och *Spara*.

En bild säger inte alltid mer än tusen ord. Idealet vore om en ikon kunde tolkas lika lätt som text på en knapp. En bra ikon talar direkt om för användaren vad det är för funktion som döljer sig därunder och är oftast effektivare än en hel mening. Om ikonerna kräver en förklaring för att förstå vad den är till för, har syftet med den gått förlorat.

Generellt sett är det enklare att beskriva ett substantiv (*skrivare*, *diskett*) med en bild, än att beskriva ett verb (*Spara*). Vid val av bild är det lämpligt att börja med de fysiska objekt som berörs, exempelvis *skrivare* för *utskrift*. Om inte detta ger en tillfredsställande beskrivning kan resultatet av händelsen övervägas. *Rensa skärmen* kan exempelvis beskrivas med en bild på tom sida. Om inte detta passar så försök hitta något enkelt sätt att beskriva händelsen ifråga. Ikoner som ska beskriva en invecklad händelse riskerar att sluta som små tecknade serier, totalt obegripliga för användaren och dessutom utrymmeskrävande.

Likväl som ikonerna måste vara lätta att tolka måste de också vara lätta att särskilja, eftersom användaren lätt ska känna igen dem och kunna avgöra vilken kontroll som gör vad. Det är inte heller lämpligt att använda liknelser, ordvitsar och annat för att beskriva en kontroll eftersom användare har en vitt skild uppfattningsförmåga och associerar på olika sätt. Många program används också internationellt och flera ord har olika betydelse beroende på språk och kultur.

Nyborjaren i ikon-design bör var försiktig med att använda för mycket färger på ikonerna. Använd ett litet antal färger, inte fler än tre eller fyra stycken förutom gråskalorna. Använd gärna färger som är lätta att skilja åt och var noga med att ge samma färg åt samma element på olika ikoner. Färgerna ska stödja bildspråket, så försök att ge bilderna så korrekta färger som möjligt. Använd starka kontraster för att avgränsa färgerna från varandra och svarta linjer för att rama in ikonerna. Kom ihåg att när ikonerna slutligen används kommer den att vara mycket liten, men den måste ändå gå att urskilja på ett ögonblick. Var också noga med att ikonerna går att särskilja utan hjälp av färgerna.

Många ikoner, speciellt de som representerar applikationer på skrivbordet eller i fönsterhanteraren är designade för att ha ett tredimensionellt utseende. Den viktigaste regeln för att få djup på bildskärmen är att objektet ska verka vara upplyst från användarens vänstra axel sett. Det här gör att under- och högersidan verkar ligga i skugga och det mänskliga ögat tolkar detta som att ikonerna står upp ifrån ytan.

Det mänskliga ögat är ganska lätt att lura, det ser gärna tre dimensioner och kurvor där inget av det egentligen existerar. Det finns en del riktlinjer för att utnyttja detta:

- Använd vita linjer för att lysa upp kanter och mörkgrå linjer (bättre än svart) för skuggor.
- Att lysa upp ovansidan och vänstersidan ger en upphöjd effekt medan undersidan och högersidan ger en nersänkt effekt. Detta kan lätt ses i tredimensionella versioner av standardkontroller som används i många applikationer.
- Linjer behöver inte nödvändigtvis vara av samma tjocklek, det vanliga är att diagonala linjer upplevs vara tunnare än vertikala och horisontella linjer, de kan därför behöva göras bredare. Linjer behöver inte heller bestå av enbart en färg, för att till exempel få diagonala linjer att verka bredare är det lämpligt att använda sig av varierande enkel och dubbel tjocklek i mörkgrått, ljusgrått och svart.

Markörer. De flesta plattformarna använder sig av olika markörer för feedback till användaren: ett timglas betyder vänta, en pil betyder klicka och så vidare. Att ändra markörens form har den stora fördelen att användaren inte kan missa det eftersom det sker där användaren har sin uppmärksamhet. Principerna för design av markörer liknar dem som används vid design av ikoner - det är dock ännu viktigare att enkelt kunna särskilja olika markörer.

Möjligheten att kunna ändra formen på markören kan användas på flera olika sätt:

- Indikera olika status, exempelvis *Vänta* med hjälp av ett timglas.

- Indikera otillåtna fält med hjälp av kryss så att användaren ser att fältet inte går att fylla i eller ändra.
- Visa när urklipps-buffern innehåller material som kan klistras in.

Markörer och även ikoner används mer och mer i animationer, till exempel som indikatorer för att visa hur en process fortskrider. Norton Desktop använder sig till exempel av en animerad filhög för att indikera sökandet efter en fil och Microsoft Word Version 6.0 använder sig av pappersark som matas genom en skrivare för att visa att utskrift pågår.

Det egentliga värdet av animeringar, förutom att användarna ofta är väldigt förtjusta i dem, är diskuterbart. Rörelser drar oftast uppmärksamhet till sig och om händelsen som framkallar animationen ska ske i bakgrunden, kan det diskuteras hur lyckat det egentligen är. I de fall då en process tar lång tid kan de vara användbara för att lugna användaren, eftersom denne då ser att någonting händer istället för att tro att datorn har hängit sig. Lämpliga animationer i detta fall kan vara *timglas* eller *nedräkning*. Speciellt användbara är dessa vid användning av Internet, eftersom nätverkskopplingar ofta kan ta lång tid och även hänga sig.

Skräddarsydda kontroller. Skräddarsydda kontroller så som specialanpassade menyer och knappar, har blivit mycket populära, delvis på grund av att de ofta är mer visuellt attraktiva för användaren men också på grund av att de med dagens utvecklingsverktyg är ganska enkla att skapa .

Det krävs dock viss försiktighet vid skapandet av skräddarsydda kontroller eftersom de kan leda till att applikationen blir svårare att använda och tar längre tid att lära då den skiljer sig från andra applikationer

Givetvis finns det tillfällen då skräddarsydda kontroller passar mycket bra att använda, speciellt när de har en stark verklighetsförankring. Ibland kan en ny kontroll leda till att en gammal kontroll ersätts, vars användning inte riktigt stämmer överens med vad som vill uppnås med applikationen. Ett bra exempel på när det är lämpligt att använda skräddarsydda kontroller är till exempel volymkontroller, där det finns mycket att vinna i förståelse hos användaren om idéer hämtas ifrån verkligheten.

Det är oftast enklare att skapa en skräddarsydd kontroll för enbart output, till exempel en termometer som visar temperatur, än vad det är att skapa en för input. Detta eftersom användaren bara behöver kunna läsa av en output-kontroll och inte behöver förstå hur nya värden läggs in eller ändras.

Innan en skräddarsydd kontroll tillverkas bör det undersökas om det finns någon standardkontroll som skulle passa. Om så är fallet, bör det noggrant övervägas vilka fördelarna är med en skräddarsydd kontroll. Om svaret enbart

är att göra kontrollen mer attraktiv utseendemässigt sett, bör beteende och utseende baseras så mycket som möjligt på standardkontrollen.

Färger

För att kunna använda färger effektivt är det viktigt att förstå hur människor uppfattar dem, hur bildskärmen återspeglar dem, samt att ha en grundläggande kunskap i färgsammansättning.

Använd färger med måtta. En bildskärm som ser ut som en upplyst julgran distraherar användaren från sitt arbete och denne kan få känslan av att inte tas på allvar. Överanvändning av färger får skärmen att se rörig ut, förvirrar användaren och ökar möjligheterna till fel.

En bra regel att hålla i minnet är att ju mindre en färg används, desto större är chansen att den får användarens uppmärksamhet när den väl utnyttjas. Om exempelvis rött används till varning, så bör det inte användas till något annat. När en färg används minimalt är effekten maximal!

Bakgrundsfärgen påverkar effekten av de övriga färgerna, som helst inte bör vara mer än fem. Välj en diskret bakgrundsfärg - undvik att använda solida, svarta omöstrade bakgrunder, eftersom detta kan få det att verka som att alla tecken på skärmen flyter omkring på olika avstånd relativt till bakgrunden. Om en svart bakgrund måste användas, så bör den kompletteras med vitt och ljusa nyanser av rött, gult, grönt och blått.

Ett bra val för bakgrund är en mönstrad matt ljusgrå färg. Denna typ av bakgrund hjälper användaren att hålla uppmärksamheten på texten och grafiken i applikationen, motverkar känslan av att tecknen flyter omkring och tillåter designern att använda sig av svarta objekt och text.

Använd färger konsekvent med användarnas förväntningar. Beroende på kultur och erfarenhet, har vi gradvis skapat oss olika uppfattningar om vad färger betyder. Vid trafikljus betyder till exempel rött *Stopp*, gult *Var uppmärksam* och grönt *Kör*. Bilens bromsljus och stoppsignaler är röda, medan bärningsbilens ljus är gula för uppmärksamhet.

Det är viktigt att överväga detta när färger används i en applikation. De vanligaste exemplen följer här nedan.

Rött: *Varning, stanna, fel, varm*
Grönt: *Kör, Okej*
Gult: *Uppmärksamhet, långsamt*
Blått: *Kallt*

Alltså, använd rött som varning för att data kommer att raderas om handlingen utförs. Använd grön text eller grafik för att tala om att den efterfrågade

processen har utförts. Använd gult när en process tar lång tid. Använd färgerna på ett sätt som användaren förväntar sig.

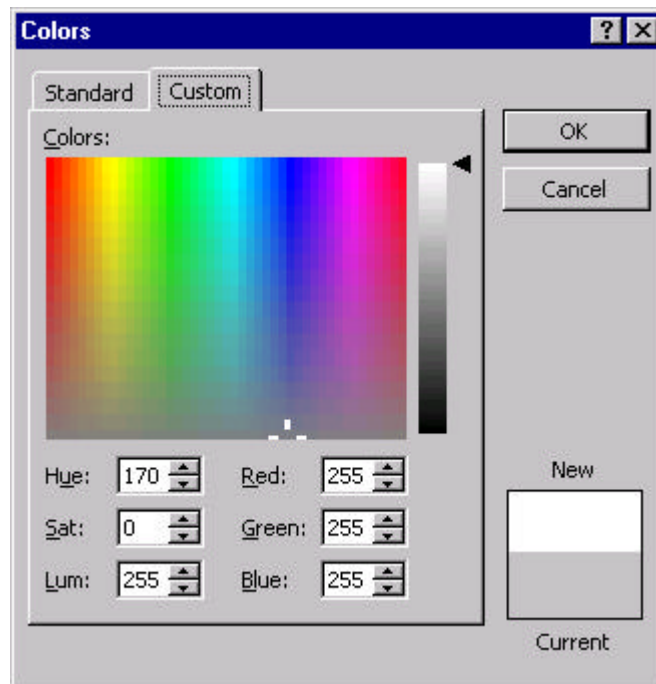
Använd färger med stor kontrast. Med kontrast menas skillnaden i ljusstyrka mellan två objekt. Ögat fokuserar på det objekt som står i störst kontrast till bakgrunden. Detta betyder att bakgrundsfärg och övriga färger måste väljas med försiktighet. Använd till exempel inte gul text mot en ljusgrå bakgrund. Den dåliga kontrasten mellan färgerna gör att det blir svårt för ögat att fokusera på bokstäverna och texten blir svårläst. Utnyttja istället opponenter som svart/vitt. Undvik blått till text och linjer, det mänskliga ögat har nämligen svårt att uppfatta detta klart.

Mättade färger, dvs väldigt klara färger, kan ge en falsk bild av djup på skärmen. Fastän de ligger på samma nivå kan de upplevas olika. Mättad röd upplevs exempelvis ligga närmare än mättad blå. Objekt med dessa mättade färger kan upplevas som att de flyter omkring framför eller bakom skärmen. Detta fenomen beror på att ögat är tvunget att fokusera om för varje mättad färg på skärmen.

Dessa färger kan även orsaka andra visuella problem. Mättade färger med samma ljusstyrka är svåra att särskilja för människor med nedsatt färgseende, där mättad röd är särskilt svår att urskilja (Human factors society 1988). Tittar användaren för länge på en mättad färg kan denne dessutom ofta se färgfläckar av den motsatta färgen en bra stund efteråt (Narbourogh-Hall 1985).

Var försiktig i användandet av mättade färger i en applikation. Blanda i lite vitt så blir färgerna mycket behagligare att titta på.

Låt användarna bekänna färg. Färgsättning är väldigt personlig. De färger som designern föredrar kan användaren mycket väl ogilla. Bästa sättet att göra detta på är att sätta diskreta färger som standardvärden och sedan tillåta användaren att ändra dessa genom att skraddarsy sina egna färger. Funktionen för detta bör vara lätt att använda. Låt till exempel användaren välja mellan ett antal befintliga färger eller låt denne skapa sina egna färger. Användaren bör då få välja ett av de visade objekten och sedan själv justera ljusstyrka, mätthet och liknande. Detta görs enklast genom använda slider-kontroller där användaren drar kontrollerna till önskad färgnyans. Därefter bör användaren antingen kunna använda sin färg eller återgå till standardfärgen.



Figur 10. Exempel på hur användaren själv kan välja färg med hjälp av en slider-kontroll.

Ytterligare en sak att tänka på vid användning av färger. I ett människa-dator gränssnitt, kan både människan och datorn vara oemottagliga för färg. För människan används ofta termen färgblind även om det för det mesta handlar om nedsatt färgseende, där problemet ligger i att urskilja vissa färger. På samma sätt kan datorer fortfarande ha monokroma skärmar och kan då inte visa de olika färgvalen.

Då det finns både människor, skärmar och skrivare som inte kan uppfatta färg bör det också användas text och ikoner för att identifiera objekt.

Sammanfattning

Att designa ett användargränssnitt kräver kunskap i layout och färglära. Alla beslut som rör detta område bör också baseras på en grundlig kunskap om användaren. Detta innefattar en förståelse av de generella styrkor och svagheter av hur människan tar emot och behandlar information, likaså en generell profil av kunskapsnivån på den avsedda användargruppen samt deras arbetsuppgifter.

Det har utvecklats många modeller om hur det mänskliga minnet fungerar. De flesta utgår ifrån att minnet består av ett kortidsminne som behandlar information med begränsning i tid och mängd, och ett långtidsminne där det finns kapacitet att lagra mycket information under ett längre tidsperspektiv.

Det är också viktigt att ta hänsyn till att kunskapen hos användarna varierar. Systemet ska vara effektivt för den erfarna användaren och lätt att lära för den oerfarna.

För att ett system ska vara attraktivt på marknaden räcker det inte med att det är funktionellt, det måste också vara estetiskt tilltalande. Med det grafiska användargränssnittet (GUI) har det introducerats en helt ny värld av form och färg, som har gjort gränssnittet mer användarvänligt, men som också ställer större krav på designern. Största försiktighet bör iakttas så att användaren inte drunknar i färgsprakande överarbetade applikationer.

Den största fördelen med GUI:s är användningen av så kallad ***direkt manipulation***. Med detta menas att operationer utförs direkt med mus eller från tangentbord och resultatet blir direkt synligt på skärmen.

För att underlätta i gränssnittsdesign har det utvecklats en del riktlinjer, både allmänna och mer konkreta vad det gäller till exempel positionering av kontroller och användning av färg. Under de allmänna riktlinjerna ges bland annat råden att hålla den underliggande teknologin osynlig, använda metaforer från verkligheten och att systemet ska vara förlåtande vid misstag. Riktlinjerna för layout ger vägledning av mer visuellt slag för fönster, verktygsfält, menyer, knappar och färgsättning. Följande punkter är exempel på vad som behandlas:

- vid fönsterdesign är det till exempel viktigt med storlek, position, form och stil
- menyer bör inte ha fler än tre nivåer
- verktygsfält är lämpliga för funktioner som används generellt och ofta
- knappar bör användas sparsamt och för mer specificerade val
- färger bör användas sparsamt och med en diskret bakgrund

INTERNET EXPLORER VS NETSCAPE NAVIGATOR

- EN STUDIE I

DESIGN OCH ANVÄNDARVÄNLIGHET

Introduktion

Allt eftersom Internet har ökat i popularitet som media för informationssökning och annonsering, samtidigt som det används till avancerade funktioner såsom databas-applikationer och interaktiva applikationer, ställs numera allt större krav på webbläsarna. De används av drygt 160 miljoner människor (NUA Internet Surveys), och måste klara av allt från nybörjarens tafatta försök till den erfarne datoranvändarens krav på effektivitet och funktionalitet.

I denna del utvärderas de två populäraste och mest avancerade webbläsarna som används idag – Internet Explorer och Netscape Navigator. Resultatet som presenteras behandlar inte varje enskild funktion i detalj, då detta ej är relevant för den inriktning vi valt. Särskilda egenskaper kommer att utvärderas, med tyngdpunkt på design och användarvänlighet och med stöd av det som framkommit i föregående del.

Syfte

Vi vill med denna fallstudie praktisera de riktlinjer som formulerats i första delen av uppsatsen för att se om de har någon förankring i verkligheten. Vissa av riktlinjerna har dock formulerats om för att anpassas till design för webben.

Utvärderingsmetoder

Det finns ett stort utbud av metoder för utvärdering av datasystem. Några där användarna är delaktiga, och andra där de helt lämnas utanför.

En traditionell metod vid tester med användare, är den som mäter om ett visst mål uppnås eller inte. Detta utförs för det mesta i ett laboratorium, där en grupp av testanvändare samlas och får utföra en mängd fördefinierade uppgifter. Under tiden samlas data om fel som görs och hur lång tid som krävs för att utföra uppgifterna. Men en metod som denna tar lång tid, kostar mycket och kräver experter, något som många systemutvecklings-projekt inte har utrymme för, varken ekonomiskt eller tidsmässigt.

Att använda sig av vanliga frågeformulär, eller att låta testanvändaren direkt under körning tala om hur denne uppfattar systemet, är ytterligare två metoder som inkluderar användarna. Dessa metoder är billigare, men har också sina nackdelar. Det är varken naturligt eller särskilt lätt för en användare att tänka

högt, och frågeformulär baserar sig på vad användaren har svarat att de tror att de gör, vilket kan skilja sig ordentligt från vad de egentligen gör.

Det finns även ett antal utvärderingsmetoder som helt utelämnar användaren, sk *inspektions metoder*. En av dessa är *kognitiv genomgång* (Polson, Lewis, Rieman & Wharton, 1992), där utvärderaren undersöker varje händelse av en lösning och försöker komma fram till varför den förväntade användaren skulle välja en viss väg genom systemet.

En annan inspektionsmetod är *den heuristiska metoden* (Nielson & Molich, 1990). Utvärderaren använder sig av en mängd riktlinjer, så kallade *heuristika*, och jämför dessa med gränssnittet. Riktlinjerna utgör en slags checklista som utvärderaren använder under arbetet. Denna metod är lätt att lära ut, ej dyr att använda och det är möjligt att hitta många problem med användbarheten i ett tidigt skede.

Den metod som används här, är en variant av den heuristiska metoden, med utgång från riktlinjerna formulerade i den första delen. Åsikterna som uttrycks i undersökningen kommer från ett antal studenter från Institutionen för Informatik, författarna av denna rapport inkluderade.

De versioner som kommer att användas är Netscape Navigator 4.5 och Internet Explorer 4.0.

Bakgrund

Det är nu ungefär fem år sedan *Mosaic*, den första webbläsaren med grafiskt användargränssnitt utvecklades. Innan webbläsarnas tid, runt 1991 utvecklade Universitetet i Minnesota *The Internet Gopher*, som delade in informationen på Internet i hierarkiska listor. Ungefär samtidigt utvecklade en man vid namn *Tim Berners-Lee* och hans kollegor vid *Cern* (European Laboratory for Particle Physics) ett nätbaserat system med hypertext som de kallade *World Wide Web*. Till skillnad från Gophers hierarkiska struktur inkluderade WWW inbäddade länkar som ledde vidare till andra dokument.

1993 annonserade NCSA (National Center for Supercomputer Applications) att de planerade att släppa en ny webbläsare, kallad *Mosaic*. Enligt många var Mosaic det verktyg som först utnyttjade webben för nätverksbaserad informationsdistribution. Fram till ungefär 1994 fanns det två huvudsakliga webbläsare - *Mosaic* och *Cello*. På senare år har ett antal efterträdare lanserats, med en distribution som skilt sig radikalt från någon annan produkt i mjukvaru-industrin.

En anledning till alla dessa efterträdare var det faktum att webbläsare i det stora hela är gratis. Eftersom de kan distribueras via nätet behövdes ingen förpackning, distribution eller återförsäljning.

Då det knappt krävs någon tid för att placera en webbläsare på marknaden kunde utvecklarna lägga till nya funktioner eller buggfixar dagligen. Denna pågående uppgraderingscykel tillät, och tillåter fortfarande, användare att enkelt ladda hem gratis kopior för utvärdering.

Mosaics historia efter 1994 har varit händelserik: ett konstant flöde av uppdateringar och licensiering av programmet till ett flertal företag, däribland Microsoft. Det har också lett till *Netscape Navigators* tillkomst. Ett tag verkade det som att *Navigator* ensam skulle dominera som webbläsare, eftersom den snabbt gick om Mosaic i funktion och snabbhet. Men så gav sig Microsoft in i spelet med *Internet Explorer 3.0*, som var en ordentlig uppdatering gentemot tidigare versioner, vilka saknade de flesta av Netscape Navigators funktioner.

Idag är Netscape Navigator och Internet Explorer de två ledande webbläsarna. Trots många likheter beroende på att de båda är gjorda för att stödja Hypertext Markup Language (HTML), finns det också stora skillnader mellan de båda.

Undersökning

Som utgångspunkt till denna undersökning, har vi använt oss av det som framkommit i föregående avsnitt. Eftersom webbläsare är specialfall av användargränssnitt, som inte har likhet med varken andra applikationer på nätet eller med vanliga informationssystem, har vi fått ställa upp ett set specialanpassade riktlinjer. Därför stämmer de inte alltid överens med de som tidigare beskrivits, men vi har försökt att hålla förändringarna nere till ett minimum. Dessutom har webbläsare attribut som ej vanliga applikationer eller webbsidor har, men som vi anser vara så viktiga att de förtjänar att omnämnas.

Eftersom användarna i det här fallet är mer än 160 miljoner människor, är det svårt att bedöma användarvänlighet utifrån en viss målgrupp. Det blir istället en allmän undersökning, med mycket betoning på den visuella layouten.

Undersökningen baseras på de engelska versionerna av webbläsarna Netscape Navigator 4.5 och Internet Explorer 4.0. På grund av detta använder vi oss av engelska termer vid beskrivning av de olika objekten. Netscape Navigator kommer att förkortas med *NN*, och Internet Explorer med *IE*.

Design

Färger, fonter, ikoner och övriga attribut skall stödja användarens arbete, inte ta uppmärksamheten ifrån den relevanta uppgiften. Hur pass väl följer då Netscape Navigator och Internet Explorer de formulerade riktlinjerna när det gäller layout?

Fönster

Storlek, position, form och stil är de viktigaste faktorerna när det gäller fönster.

- Hur utnyttjas fönsterytan?

IE:

- + När vissa funktioner väljs visas resultatet i en sidofönster, så att det går att arbeta med den informationen samtidigt som den aktuella sajten är synlig.
- + Det finns möjlighet att välja funktionen *Full screen* för att utnyttja skärmytan bättre, menyraden visar då bara det nödvändigaste på en liten yta längst upp.
- Om det är en liten skärm så tar sidomenyerna upp för mycket av skärmytan, vilket lätt kan bli en störande faktor.
- Huvudfönstret öppnas inte i maximalt läge, vilket orsakar ett extra moment, då användaren normalt arbetar med webbläsaren i helskärm.

NN:

- + När ett fönster öppnas, visas det direkt under aktiveringskontrollen.
- + Huvudfönstret kan göras större genom att minimera eller ta bort verktygsfälten.

Verktygsfält/Paletter

Syftet med verktygsfält är att ge snabb tillgång till vanliga händelser. Endast de funktioner som används frekvent bör finnas där.

- Är verktygsfälten lämpligt placerade?
- Är objekten som visas i verktygsfälten relevanta?

IE:

- + Verktygsfälten kan med hjälp av musen förflyttas praktiskt taget vart som helst på skärmen.
- + Lätt att anpassa verktygsfälten i och med att de kan plockas bort och läggas till efter användarens behov.
- + *Favourites*, *History*, *Full screen*, *Mail* och *Print* är placerade som lättåtkomliga ikoner på ett av verktygsfälten.
- I och med att verktygsfälten är så justerbara, är det lätt att tappa bort både sig själv och funktionerna.
- Vad gör *Edit*?

NN:

- + Verktögsfälten är indelade i tre kategorier: navigering, lokalisering och ett personligt fält. Dessa går att dölja via minimering och de går även att plocka bort helt så att funktionerna endast går att nå via menyn.
- + För den erfarna användaren är det personliga verktögsfältet ett bra inslag. Här kan genvägar placeras efter eget behag, vilket leder till effektivare användning av webbläsaren.
- Paletten för mail- och composerfunktionerna ligger lite undandömd i det nedre högra hörnet. Användaren kan låta paletten bli till en flytande komponent, men detta är bara ett störande inslag.

Menyer

För att underlätta vid navigation i systemet placeras ofta kommandon och händelser i menyer. De bör placeras i logisk ordning och ej innehålla för många nivåer.

- Är menyobjekten lämpligt grupperade?
- Är det lätt att följa nivåuppbbyggnaden?

IE:

- + har använt sig av menyer med windows-standard vilket gör att de flesta användare lätt känner igen sig.
- + Menyerna är korta och koncisa.
- + Verktögsfälten bra grupperade under *View*.
- + Bra indelning av bokmärken.
- Vissa menyer har upp till fyra nivåer, men detta verkar mer vara ett undantag än standard.

NN:

- + Menyn är uppbyggd efter Windows standard, vilket gör det lätt för användare att känna igen sig. Funktioner som *Skriv ut* och *Kopiera* ligger på samma ställe som i Windows applikationer.
- Under menyn "*Communicator*" har många funktioner förts samman, som inte riktigt logiskt hör ihop. Verktögsalternativen känns rätt malplacerade här. Borde kanske istället ha en egen meny.
- "*Search Internet*" ligger under "*Edit*", men det borde logiskt sett istället placeras under till exempel "*Go*".
- Vissa menyval har upp till fyra nivåer, vilket är på gränsen till vad en användare klarar av att följa.

Kontroller

Kontroller är till för användarens interaktion med datorn. De används både för input och output, där de kan användas antingen för att ta fram befintliga värden eller för att lägga till nya.

- Går det via utseendet på kontrollen att bedöma vad den skall användas till?

- Är kontrollen relevant för uppgiften?

IE:

- + Bra verklighetsförankrade metaforer som ikoner. Parabolen som representerar *Channels* är ett mycket bra exempel på en passande metafor.
- + Bra och tydlig ikon för mail-funktionen.
- + Bra med kompletterande text under ikonerna.
- Antal ikoner som visas är många, och det känns lite ”plottrigt” på verktygsfältet. Det ställer även till problem om skärmen är av modell mindre.

NN:

- + Metaforerna visar tydligt den underliggande funktionen. Pilar för förflyttning, ficklampa för sökning och trafikljus som visar rött för stoppfunktionen är bra och talande exempel.
- + Bra med förklarande text under ikonerna.
- + När musen dras över ikonerna visas en förklarande text till vad funktionen utför.
- Några kontroller har en undanskynd position vid nedre kanten av fönstret.
- Kontrollen för säkerhet visas på två olika positioner på fönstret, och finns dessutom som menyval.

Färger

Färger bör användas konsekvent och ej överdrivas. Överanvändning leder till förvirring och distraherar användaren.

- Förstärker färgerna applikationens funktion?

IE:

- + Diskret och verklighetsanpassad färgsättning.
- Ikonerna är grå tills dess att fokus sätts på dem, detta gör dem svåra att särskilja.

NN:

- + Den grå bakgrunden bidrar till att hålla webbläsaren i bakgrunden så att fokus sätts på den relevanta informationen i fönstret.
- + Ikonerna har en diskret framtoning med blåa inslag, vilket är vilsamt för det mänskliga ögat.
- + Färgerna används utifrån användarnas förväntningar, till exempel rött stoppljus och gröna pilar för förflyttning.
- Ikonerna kunde kanske ha skilt sig åt lite mer färgmässigt, för en tydligare framtoning

Användarvänlighet

Användargränssnitt är till för att underlätta för användaren att lösa sina uppgifter. Därför måste de designas på ett så funktionellt och förklarande sätt som möjligt. Nedan undersöks webbläsarnas användarvänlighet.

Systemstatusens framtoning

Det viktigaste en användare behöver veta vid besök på webben, är "*Var befinner jag mig?*", "*Vad sker just nu?*" och "*Hur fortsätter jag?*". Webbläsaren bör klart och tydligt tala om användarens position. Länkar till tidigare besökta sidor bör vara lättåtkomliga.

- Framgår tydligt den aktuella positionen?
- Är det lätt att komma åt tidigare besökta sidor?
- Syns det tydligt när systemet arbetar?

IE:

- + Sidrubriken visas tydligt uppe i vänstra hörnet.
- + Tidigare besökta sidor nås mycket lätt antingen via *Back*-ikonen, adressfältet eller den finurliga funktionen *History*, där sajterna besökta en längre tid tillbaka finns tillgängliga.
- + Vid nedladdning rör sig Internet Explorers ikon, en status indikator och förklarande text visas, som talar om att nedladdning sker och när denna är klar.
- Ikonernas rörelser syns lite otydligt.

NN:

- + Sidrubriken visas tydligt uppe i vänstra hörnet.
- + Tidigare besökta sidor under sessionen kan nås på olika sätt - genom att använda sig av *Back*-knappen, genom att klicka på combo-boxen som visar adressen, eller via menyvalet "*Go*".
- + Netscape Navigators logotyp är i rörelse när den söker och laddar hem material. Detta visas också med förklarande text och en visuell indikator, i den undre kanten på fönstret.
- Om användaren vill nå sidor från ett tidigare tillfälle, krävs viss kunskap om webbläsaren, eftersom denna information ligger väl dold under ett inte alltför självklart menyval.
- Vid nedladdning av en sajt flimrar texten som visar statusen så fort att det inte går att uppfatta vad som står.

Framhäv viktig information

Använd fonter, storlek på tecken, färger, former och formulär för att framhäva viktig information. Visa uppgiftsrelaterade verktyg med liknande färger och teckenstorlek och genom att använda speciella positioner för specifika

funktioner. Det kommer troligen att minska tiden som en erfaren användare behöver för att lokalisera information.

- Framhävs den viktiga informationen?
- Är den framhävda informationen viktig?

IE:

- + Det finns en Mail-ikon tydligt placerad i ett av verktygsfälten.
- + Ingen otydlig information i form av små ikoner i nedre delen av fönstret.
- Genväg till hjälpen saknas vilket gör att användaren måste gå via menyn, detta tar längre tid och gör att hjälpen blir svårare att hitta.
- Att visa *Edit* känns ganska onödigt.

NN:

- + Tydliga ikoner för viktiga funktioner som exempelvis *Search*, *Stop* och *Home*.
- Ikonen för *Security* är framhävvd trots att den knappast är viktig för den dagliga användningen av webbläsaren.
- Funktionen för e-mail syns knappt nere i hörnet, borde egentligen ligga som en ikon uppe bland verktygsfälten.
- Vissa genvägar som inte är relevanta för alla användare är satta som standardvärden på en väldigt synlig position, exempelvis *Lookup* och *New & Cool*.
- Genväg till hjälpen saknas vilket gör att användaren måste gå via menyn, detta tar längre tid och gör att hjälpen blir svårare att hitta.

Användning av genvägar och bokmärken

Genvägar skall användas för att minimera användarens input. På webbläsare finns de i form av ikoner, bokmärken och länkar.

- Används genvägarna för relevanta och ofta utförda operationer?
- Är det enkelt att skapa bokmärken?

IE:

- + När bokmärken skapas får användaren själv välja namn, placering och om denne vill bli informerad varje gång sidan uppdateras.
- + Egna länkar kan placeras på verktygsfältet.
- Det är svårt att hitta funktionen *Add bokmark*, borde kanske ligga som en ikon i verktygsfältet.
- Microsoft har själva lagt till länkar som placerats på verktygsfältet som kanske inte känns relevanta för flertalet användare.
- Edit-funktionen är nog inte en funktion som används tillräckligt ofta för att den ska ligga som en ikon i verktygsfältet.

NN:

- + Det är enkelt att skapa bokmärken.
- + Funktionen *What's related* visar sidor som har kopplingar till den aktuella sidan.
- + *Print* är satt som en ikon vilken öppnar ett fönster där utskriftsformat med mera kan väljas. Detta är en fördel då de flesta användare vill kunna styra utskriftsformatet själv.
- *Security* är satt som genväg trots att den ej utnyttjas speciellt mycket i den dagliga användningen.
- Det går inte att själv sätta namn på bokmärkena.
- Möjlighet finns att dra ikonerna som visas bredvid URL-adressen till det personliga verktygsfältet så att en genväg skapas. Detta är visserligen positivt, men det vore ännu bättre om bokmärket gick att döpa, då de alla ser likadana ut.

Användning av standardvärden

Standardvärden sätts mest för att förenkla för den ovane användaren, så att det första som händer ej är att denne behöver ändra i inställningarna.

- Är den lämpliga informationen satt som standardvärde?
- Är det lätt att ändra standardvärden?

IE:

- + Börjar användaren att skriva en redan besökt adress, fylls resten i automatiskt.
- Sökmotorn under *Search* lämnar mycket övrigt att önska.
- Det ligger bara amerikanska kanaler som standardvärden.
- Microsoft har lagt in extremt många länkar och annat krafs som tar mycket plats och kan förvirra en ovan användare.

NN:

- + Det är enkelt att ändra standardvärden så som färger på länkar, hemsida och antal dagar som länkar till besökta sidor ska sparas.
- + Under *Search* ligger NN:s egen sökmotor som håller hög klass.
- + Börjar användaren att skriva en redan besökt adress, fylls resten i automatiskt.
- Det finns många bra sökmotorer, så valet borde finnas att ändra standardvärdet under *Search*.
- Genvägar av typen *Lookup* och *New & Cool* är satta som standardvärden i det personliga verktygsfältet, trots att dessa inte används av flertalet användare. Dessutom kanske inte användaren delar uppfattning om vad som är *New* och vad som är *Cool* med utvecklarna av webbläsaren.

Språkanvändning

Systemkoncept bör uttryckas i användarens språk med ord, fraser, symboler, ikoner etc, som kan förstås av de flesta användare. Undvik systemorienterade termer.

- Är språket uppenbart för användaren?
- Är termerna svåra att förstå?
- Är det uppenbart hur uppgiften ska utföras?

IE:

- + Ledig användning av språket, inga alltför tekniska termer.

NN:

- + Termerna som används är vanliga standardtermer, vilket underlättar för användaren.
- + Enkelt att använda om grundidén med webbläsaren följs – att förflytta sig mellan olika sajter på nätet.
- + Rubriken vid adressfältet ändras från *Location* till *Go to* när en ny sajt laddas ner.
- Det används en del facktermer som kan vara svåra att förstå av mindre teknik-orienterade användare.

Återhämtningsmöjligheter

Användare väljer ofta systemfunktioner av misstag och behöver därför klart markerade ”nödutgångar” för att kunna ta sig ur oönskade situationer, utan att behöva ta sig igenom en massa onödig dialog.

- Vad finns det för möjlighet att ångra och upprepa?

IE:

- + tydliga ikoner för att stanna, backa och ladda om sidan – *Stop*, *Back* och *Refresh*.

NN:

- + tydliga ikoner för att stanna, backa och ladda om sidan – *Stop*, *Back* och *Reload*.

Felmeddelanden

Fel kommer att uppstå, trots alla ansträngningar för att undvika dem. Varje felmeddelande bör erbjuda någon slags lösning, eller en länk till en lösning på problemet. Om en användares sökning till exempel inte leder till någon träff, räcker det inte med att bara tala om för denne att utvidga sin sökning, utan denne bör förses med en länk som automatiskt vidgar sökningen.

- Innehåller felmeddelandena tillräckligt med information och är de tydliga?
- Är den relevant?

IE:

- + Felmeddelande visas i form av en liten ruta som enkelt går att klicka bort.
- Om sajten som söks inte finns ges ändå meddelandet att det ej går att kontakta servern. Tvetydigt.

NN:

- + Tydliga felmeddelanden när adressen ej existerar eller är felstavad.
- När den önskade sajten ej finns, ges onödig information om nummer på felet. Vad hjälper det användaren?

Förse användaren med hjälp och dokumentation

Vad det gäller webben så räcker det inte med att bara lägga upp några hjälpsidor, utan försök integrera dokumentationen direkt på sidorna, eller se till att det finns länkar från huvudsektionerna till direkt hjälp och vice versa. På så sätt känner användaren att hjälpen aldrig är långt borta.

- Är hjälpen lätt att hitta?
- Är den relevant?

IE:

- + Windowsstandard på hjälpen, enkelt för de flesta användare att förstå.
- + Hjälpmenyn är kort och koncis.
- Det finns ingen ikon för hjälpfunktionerna.

NN:

- + Hjälpen är mycket innehållsrik och lätt att hitta och använda.
- Mycket onödig information under hjälpmenyn.
- Det finns ingen ikon för hjälpfunktionerna.

Resultat

I denna undersökning har det, utifrån studenter på informatiks synvinkel, gjorts en bedömning av fördelar och nackdelar i design och användarvänlighet hos de två största webbläsarna idag. Trots stora likheter finns ändå en hel del skillnader i uppbyggnad och det sätt som har valts för att presentera de olika funktionerna .

Det är naturligtvis stor skillnad på olika användares åsikter om vad som är bra och dåligt i en applikation, det vissa tycker är en stor fördel tycker andra inte om över huvudtaget. Vi har med vissa modifikationer följt de riktlinjer som beskrivs i första avsnittet, och som ses som en generell standard på hur ett bra användargränssnitt ska se ut. Naturligtvis är undersökningen till viss del färgad av egna åsikter och erfarenheter av de två webbläsarna.

Design

Eftersom webbläsare fungerar som en oföränderlig ram runt den aktuella sajten, är det svårt att bedöma dem som vanliga applikationer vad gäller fönsterattribut. Det som dock kan sägas, är att det är en fördel att de fönster som öppnas i webbläsaren, placeras direkt under den funktion som de aktiveras av. I de båda miljöerna går utrymmet för själva webbsidorna att göra större genom att plocka bort verktygsfält. I Internet Explorer finns dessutom möjlighet att välja *Full Screen*, som minimerar verktygsfälten och ger webbsidorna maximalt med utrymme.

Då webbläsare är till för att förflytta sig mellan olika sajter på nätet, är det en självklarhet att användaren på ett snabbt och enkelt sätt skall kunna utnyttja funktioner såsom *Back*, *Forward*, *Home* och *Stop*. Både Internet Explorer och Netscape Navigator har anammat detta med tydliga ikoner, placerade på en visuellt utmärkande plats bland verktygsfälten. Det har också placerats andra funktioner här vars betydelse ibland kan ifrågasättas. Verktygsfälten kan i båda fallen även fungera som paletter som kan flyttas runt och kopplas loss till självständiga enheter. Detta kan dock ställa till med förtret, eftersom ett flytande objekt är svårare att hålla ordning på än en fast verktygsrad.

Båda webbläsarna har byggt upp sina menyer enligt Windowsstandard, vilket underlättar mycket för användarna. De kan direkt känna igen sig, och vet på ett ungefär var de hittar de olika funktionerna. Netscape Navigator har dock gjort vissa mindre bra menyval, då de har grupperat vissa funktioner som logiskt sett inte har mycket gemensamt. Fyra menynivåer återfinns hos båda webbläsarna, vilket är maximum för vad en användare klarar av. Internet Explorer håller dock sina menyer lite kortare och mer koncisa, vilket ger ett mer "städat" intryck.

Kontroller används flitigt i både Netscape Navigator och Internet Explorer i form av ikoner och combo-boxar. Ikonerna är representerade med verklighetsförankrade och tydliga metaforer, såsom en skrivare för *Print* och ett hus för *Home*, som används i båda fallen, kompletterade med förklarande text. I Netscape Navigator är det ännu tydligare markerat vilken den underliggande funktionen är, då det visas ytterligare förklaring när muspekaren rör vid objektet.

Båda webbläsarna använder sig av en diskret grå bakgrund med svart text, helt enligt riktlinjerna. Ikonerna visas i färger som stämmer överens med verkligheten, och mättade färger har undvikits i största möjliga mån. Internet Explorer går dock lite till överdrift i sin strävan att hålla en diskret framtoning, då de låter alla ikoner vara grå tills dess att de sätts i fokus.

Användarvänlighet

Både Internet Explorer och Netscape Navigator har valt att placera sidrubrik högst upp i det vänstra hörnet, vilket är ett praktiskt och logiskt val. Med tanke på att det mänskliga minnet har lättare att känna igen än att komma ihåg, har möjligheten att kunna gå tillbaka till tidigare besöka sajter, utan att åter behöva skriva in adressen, en viktig funktion. De båda webbläsarna har löst detta genom att placera en Back-funktion på verktygsfältet samt visa länkar i en lista under adressfältet. När det gäller att nå sajter som besökts vid en tidigare session, har Internet Explorer en fördel, då det via en ikon på verktygsfältet – *History*, enkelt går att få upp ett sidofönster som innehåller all nödvändig information och länkar. Netscape Navigator har valt att placera denna information i en undermeny, som inte är helt lätt att hitta. Vid nedladdning av sajter, sätts de två webbläsarnas logotyper i rörelse, samt att det visas en statusindikator och en beskrivande text, som talar om var som sker. Detta håller användaren informerad på ett enkelt, men funktionellt sätt.

De viktigaste funktionerna för en webbläsare är de som gör att användaren kan förflytta sig mellan olika sajter på nätet samt att kunna skicka e-mail. Det är också dessa funktioner som framhävs i de båda webbläsarna. Netscape Navigator har valt att lägga verktygsfältet för *E-mail*, *Composer* och *Discussion Groups* i en palett i nedre kanten på webbläsaren, vilket gör att den kan vara svår att upptäcka för den ovane användaren. Internet Explorer har däremot en tydlig ikon för mailfunktionen som nog är att föredra. De båda webbläsarna har i vissa fall valt att framhäva olika funktioner vilkas nytta ibland kan ifrågasättas. Många av dem kan vara svåra att förstå för användarna, och används således inte. Däremot har ingen av dem hjälpen upplagd som genväg, vilket gör att användaren alltid måste gå den längre vägen via menyn.

Att skapa bokmärken är enklare att göra i Netscape Navigator än i Internet Explorer, men den senare har fördelen att användaren själv kan döpa bokmärkena och att denne direkt får frågan om vilken mapp de ska placeras i. I Netscape Navigator sker detta automatiskt och det ges ingen bekräftelse på att bokmärket skapats. Detta kan leda till att användaren tror att aktiviteten misslyckats och då lägger in bokmärket igen. Netscape Navigator har även en funktion – *What's related* som visar relaterade länkar till den aktuella sajten. Denna funktion har dock gått många omärkt förbi, troligtvis på grund av sin undanskymda placering.

Standardvärden är något som leverantören av webbläsaren lägger in för att underlätta för användaren och ge denne någonstans att starta från. Detta kan ibland gå till överdrift, Microsoft har till exempel lagt in en uppsjö av vad de anser vara intressanta länkar som tar mycket plats och lätt kan förvirra användaren. De flesta människor har dock olika åsikter om vad som är intressant. Som tur är finns möjligheten att plocka bort och lägga till länkar efter eget behag. De båda webbläsarna har sina egna sökmotorer liggande under

Search, vilka kanske inte alltid används då det finns ett antal bättre och mer smidiga alternativ att välja mellan.

För att alla ska ha en rimlig chans att utnyttja webbläsarna fullt ut krävs det att dessa använder sig av ett språk som är naturligt för de flesta människor. Eftersom användarna kommer från alla delar av världen är detta inte en helt lätt uppgift, då olika kulturer innebär olika uppfattningar och tankesätt. De båda webbläsarna finns på många olika språk, men den amerikanska varianten är den mest använda. Microsoft har lyckats bäst i sin språkanvändning med en ledig användning av fraser och ikoner som ligger nära verkligheten. I Netscape Navigators version har det smugit sig in en del facktermer som inte är helt lätta att förstå för den vanliga användaren. I det stora hela används dock standardtermer som användaren kan känna igen från andra applikationer.

Webbläsare lämnar inte så stort utrymme för användaren att göra misstag eftersom input från tangentbordet är minimal. Det som kan ske är användaren skriver in fel adress, att sajten inte existerar eller att nedladdningen tar så lång tid att användaren vill avbryta den pågående aktiviteten. Funktionerna för att ångra, upprepa och avbryta är i dessa fall användbara funktioner och som tydligt stöds av både Netscape Navigator och Internet Explorer. De har också felmeddelanden som visas när uppgiften inte går att utföra. Dessa är dock inte alltid så väl genomtänkta och innehåller ibland onödig eller tvetydig information.

Då webbläsare egentligen inte har något att göra med sajternas innehåll så är hjälpen något begränsad. Trots detta har både Netscape Navigator och Internet Explorer lyckats skapa en hjälpfunktion som är enkel att förstå och innehåller all nödvändig information. En nackdel för Netscape Navigator är att det ligger alldeles för mycket onödig information under hjälpmenyn, vilket kan förvirra en ovan användare.

Det som slutligen kan sägas är att webbläsare är till för att förflytta sig mellan olika sajter på nätet och skicka mail på ett så enkelt och effektivt sätt som möjligt, och detta uppfyller både Netscape Navigator och Internet Explorer på ett tillfredsställande sätt. Internet Explorer har dock fördelen av att vara integrerat med Windows, då många användare känner igen både gränssnitt och design. Att överarbeta webbläsaren genom att lägga in avancerade och "flashiga" funktioner är bortkastat eftersom de flesta användare inte förstår eller har intresse att lära sig vad de är till för. Detta leder till att de sällan eller aldrig används.

DISKUSSION

Vad vi hade kunnat göra annorlunda

Att välja två webbläsare som undersökningsobjekt var kanske inte det mest uppenbara alternativet. Eftersom webbdesign är relativt nytt och skiljer sig något från design av vanliga applikationer, var vi tvungna att till viss del anpassa riktlinjerna så att de skulle stämma överens med vår undersökning. Detta försvårade vårt arbete eftersom den röda tråden emellanåt kändes tunn och vissa svårigheter med att rättfärdiga kopplingar mellan de båda delarna uppstod. Men i slutresultatet visade det sig dock att våra farhågor var orgrundade, då de båda delarna på ett naturligt sätt kopplades ihop och vi såg helheten i rapporten.

En annan punkt är att vi möjligen kunde utökat vår kategori av deltagare i vår undersökning. Det urval vi nu presenterat är inte helt överensstämmande med verkligheten, då studenter vid informatik generellt har en bättre kunskap i användning av datasystem och applikationer än gemene man. Om en bredare målgrupp använts i vår undersökning hade vi troligen fått ett större spektrum av olika åsikter rörande webbläsarnas uppbyggnad. Å andra sidan är uppsatsen ämnad för just studenter vid informatik och dessa kanske inte är intresserade av vad Greta Svensson i Gävle, som enbart använder datorn för att sköta sin hemekonomi, tycker och tänker om webbdesign.

Vad vi tycker och tänker om ämnet

Dagens webbläsare är skapade för att på ett så enkelt och effektivt sätt som möjligt kunna förflytta sig mellan olika sajter på nätet. Men tack vare den hårda konkurrens som råder idag, trappas utvecklingen av komplicerade och "flashiga" funktioner upp som inte alltid har koppling till uppgiften, och inte alltid gynnar användarna. Dessa har ej så stora förväntningar på webbläsaren, förutom en snabb och enkel sökning, men möts av en uppsjö av redan inlagda länkar, komplicerade menystrukturer och annan överflödig information.

I vissa fall känns webbläsarna överarbetade, med små, plottriga ikoner runt hela fönstret, som kan vara svåra att upptäcka och som påtvingar användaren onödigt arbete, då det krävs mycket förflyttning av musen. Tänk på att bespara användaren från musarm!

I början av arbetet var vi personligen för Netscape Navigator, då det är denna som används som standard på institutionen. Men under arbetets gång har vi hittat fördelar med Internet Explorer som gjort att vår uppfattning ändrats till viss del. Bland annat är det här lättare att hitta mail-funktionen, och dessutom har de löst problemet med att komma åt tidigare besökta länkar på ett smidigt sätt genom att öppna ett sidofönster med all nödvändig information.

Vilken webbläsare som till slut väljs beror till stor del på vilken användaren först har kommit i kontakt med, och vilken som är standard på arbetsplatsen. Skillnaden mellan de webbläsare som finns på marknaden idag, är inte stor varken i form av funktionalitet eller design, och är sällan det som avgör valet.

Vad vi kan förvänta oss i framtiden

Vi tror att det kommer att bli allt mer vanligt med möjligheten att själv kunna skraddarsy sin webbläsare. Det finns redan konkreta tendenser till detta i form av så kallade *skins*, som läggs ovanpå webbläsaren för att ge den ett personligare utseende. Vi har testat en variant, som Neoplanet har lanserat, där det går att välja mellan en uppsjö av utseenden och teman. Det gör navigering på nätet roligare, men tillför annars inte någon större funktionalitet.

En stor fråga är hur mycket de två giganterna på detta område - Internet Explorer och Netscape Navigator egentligen tar hänsyn till den enskilda användaren eller om det mest handlar om att mäta krafter sinsemellan. I vilket fall som helst är ju detta till fördel för oss användare eftersom utvecklingen hela tiden rör sig framåt. Men med detta följer också vissa problem. Ständiga uppdateringar och ändringar leder till förvirring och i värsta fall problem med kompatibiliteten.

Som det ser ut idag, styr Internet Explorer och Netscape Navigator i stor del vad som kan visas på nätet, tack vare att det är de som sätter standards. Från början var det tänkt att webbläsarna skulle göra Internet tillgängligt för alla, men det verkar nu som att det blir de som kommer att sätta restriktioner för vilket material som går att nå, beroende på vilket format de stödjer.

REFERENSLISTA

Litteratur

b) **Bickford Peter**, Interface Design - the Art of Developing Easy to Use Software, Boston AP Professional cop, 1997

Brown C. Marlin, Human-Computer Interface Design Guidelines, Norwood, N. J. Ablex Pub.Corp cop, 1988

a) **Lif Magnus**, Adding Usability: methods for modelling, user interface design and evaluation, Uppsala Universitet 1998

Löwgren Jonas, Human Computer Interaction, Lund Studentlitteratur, 1993

Mayhew Deborah J, Principels and guidelines in Software user interface design, Englewood Cliffs Prentice Hall, 1992

Nielsen J, Molich R, Heuristic evaluation of user interfaces In J.C.Chew, & J. Whiteside (Eds.), Proceedings of Human Factors in Computing Systems, 1990

Polson P.G, Lewis C, Rieman J, & Wharton C, CognitiveWalkthroughs: A Method for Theory-Based Evaluation of User Interfaces, International Journal of Man-Machine Studies, 1992

b) **Sommerville Ian**, Software Engineering, Fifth Edition, Wokingham Addison-Wesley cop, 1995

b) **Zetie Carl**, Practical User Interface Design, London, McGraw-Hill, 1995

Artiklar

Head Alison J, A question of interface design: How do online service GUIs measure up?, Online nr 21, 1997

Joch Alan, Interface The Facts, Inc nr19, 1997

Lidfeldt Torun, Användbarhet – gör gränssnittet till din vän, Datateknik nr 29, 1998

Van Dam Andries, Post-WIMP User Interfaces, Association for Computing Machinery nr 40, 1997

Internet

Gargaro Carolyn C, Netscape Navigator Vs. Microsoft Internet Explorer - Who Will Win The "Browser Wars," and Will The Outcome Affect Desktop Computing?, 8 december 1996

<http://scis.nova.edu/~gargaro/MMIS626/browsers.html>

a) **Instone Keith**, Usability Heuristics for The Web, maj 1999

<http://webreview.com/wr/pub/97/10/10/usability/sidebar.html>

Neoplanet, maj 1999

<http://www.neoplanet.com>

Nua Internet Surveys, maj 1999

http://www.nua.ie/surveys/how_many_online/index.html

Piaget Jean, Stage theory, 1969

<http://www2.hihm.no/Rena/hjemmesider/JohnHarald/jeanpiaget/index.html>

a) **Togazini Bruce**, Ask Tog – The designer is in, november 1998

<http://www.asktog.com/basics/firstPrinciples.html>

a) och b) är hänvisningar från fotnoter